



EM35x

RF4CE Application Framework Reference:

For EmberZNet 5.7.3

July 1, 2016
AF_RF4CE – 5730

Silicon Laboratories Inc.
400 West Cesar Chavez
Austin, TX 78701
Tel:1+(512) 416-8500
Fax:1+(512) 416-9669
Toll Free:1+(877) 444-3032
www.silabs.com



Disclaimer

The information in this document is believed to be accurate in all respects at the time of publication but is subject to change without notice. Silicon Laboratories assumes no responsibility for errors or omissions, and disclaims responsibility for the functioning of undescribed features or parameters. Silicon Laboratories makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Silicon Laboratories assume any liability arising out of hte application or use or any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. Silicon Laboratories products are not designed, intended, or authorized for use in applications intended to support or sustain life, or for any other application in which the failure of the Silicon Laboratories product could create a situation where personal injury or death may occur. Should Buyer purchase or use Silicon Laboratories products for any such unintended or unauthorized application, Buyer shall indemnify and hold Silicon Laboratories harmless against all claims and damages.

Silicon Laboratories, Silicon Labs, and Ember are trademarks of Silicon Laboratories Inc.

Other products or brandnames mentioned herin are trademarks or registered trademarks of their respective holders.

About This Guide

Purpose

This document is a unified collection of API reference documentation covering EmberZNet PRO Stack.

Silicon Labs recommends that you use this document as a searchable reference. It includes all of the information contained in the html version of these materials that are provided as an online reference for developers of EmberZNet-based ZigBee wireless applications. There are three key advantages that this document provides over the online html versions:

- Everything is contained in this single document.
- This document is fully searchable using the Adobe Acrobat search engine that is part of the free Acrobat Reader (available from www.adobe.com).
- This document can be easily printed.

Audience

This document is intended for use by programmers and designers developing ZigBee wireless networking products based on the EmberZNet PRO Stack Software. This document assumes that the reader has a solid understanding of embedded systems design and programming in the C language. Experience with networking and radio frequency systems is useful but not expected.

Getting Help

Development kit customers are eligible for training and technical support. You can use the Silicon Labs web site www.silabs.com/zigbee to obtain information about all Ember products and services.

You can also contact customer support at www.silabs.com/zigbee-support.html.

Chapter 1

Introduction

The Ember ZigBee Radio Frequency for Consumer Electronics (RF4CE) Application Framework provides customers with robust implementations of RF4CE standards. Furthermore, it enables users to build custom applications on top of these implementations using application framework entities such as, but not limited to, callbacks (see [Ember RF4CE Application Framework Callbacks](#)), a command line interface, sample applications, and architecture-agnostic plugins.

This API reference includes sections on the following.

- [Ember ZigBee RF4CE API Reference](#)
- [Ember RF4CE Application Framework Callbacks](#)

Chapter 2

Deprecated List

Global `emberAfRf4ceZrc20ActionMappingClientLookUpActionMapping` (`uint8_t pairingIndex, EmberAfRf4ceDeviceType actionDeviceType, EmberAfRf4ceZrcActionBank actionBank, EmberAfRf4ceZrcActionCode actionCode, EmberAfRf4ceZrcActionMapping *actionMapping`)
This function is deprecated and will be removed in a future release. Customers should use `emberAfRf4ceZrc20ActionMappingClientGetActionMapping` instead.

Chapter 3

Module Index

3.1 Modules

Here is a list of all modules:

Ember ZigBee RF4CE API Reference	12
Generic Device Profile	13
Generic Device Profile Identification Client	24
Generic Device Profile Identification Server	25
RF4CE Multiple System Operators Profile	26
RF4CE Multiple System Operators IR-RF Database Originator	34
RF4CE Multiple System Operators IR-RF Database Recipient	36
RF4CE Profile Support	38
ZigBee Remote Control 1.1 Profile	51
ZigBee Remote Control 2.0 Profile	55
ZigBee Remote Control 2.0 Action Mapping Client	61
ZigBee Remote Control 2.0 Action Mapping Server	65
ZigBee Remote Control 2.0 Home Automation Client	67
ZigBee Remote Control 2.0 Home Automation Server	86
Ember RF4CE Application Framework Callbacks	103
Framework Callbacks	104
bulb-pwm-driver Callbacks	105
Button Callbacks	107
Debug Basic Library Callbacks	108
EZSP Common Callbacks	109
HAL Library Callbacks	117
Idle/Sleep Callbacks	118
Low Voltage Shutdown Callbacks	120
Main Callbacks	121
Microphone Codec MSADPCM Callbacks	123
Microphone IMAADPCM Callbacks	124
Generic Device Profile Callbacks	125
RF4CE Multiple System Operators Profile Callbacks	129
RF4CE Profile Support Callbacks	132
RF4CE Stack Library Callbacks	148
RF4CE Target Communication. Callbacks	154
ZigBee Remote Control 1.1 Profile Callbacks	155
ZigBee Remote Control 2.0 Profile Callbacks	157

Stack Minimal Library Callbacks	163
STM32F103RET Library Callbacks	165
sim-eeprom API Callbacks	166

Chapter 4

Data Structure Index

4.1 Data Structures

Here are the data structures with brief descriptions:

DestStruct	168
EmAfBindingInfo	168
EmAfDiscoveryOrPairRequestData	169
EmAfGdpPairingCandidat	170
EmAfMsoPairingCandidat	171
EmAfRf4ceGdpAttributeDescriptor	173
EmAfRf4ceGdpAttributes	173
EmAfRf4ceMsoAttributeDescriptor	175
EmAfRf4ceMsoPeripheralIdEntry	176
EmAfRf4ceMsoRibAttributes	176
EmAfRf4cePowerSavingState	178
EmAfRf4ceZrcAttributeDescriptor	178
EmAfRf4ceZrcAttributes	179
EmAfZrcArrayedBitmask	180
EmAfZrcBitmask	180
EmberAfRf4ceGdpAttributeIdentificationRecord	
RF4CE GDP attribute identification record for Get Attributes and Pull Attributes messages	181
EmberAfRf4ceGdpAttributeRecord	
RF4CE GDP attribute identification record for Set Attributes and Push Attributes messages	182
EmberAfRf4ceGdpAttributeStatusRecord	
RF4CE GDP attribute identification record for Get Attributes Response and Pull Attributes Response messages	182
EmberAfRf4ceGdpRand	
This data structure contains the GDP random byte string that is passed into various other functions	183
EmberAfRf4ceGdpTag	
This data structure contains the GDP tag value that is passed into various other functions	184
EmberAfRf4ceMsoIrRfDatabaseEntry	
RF4CE MSO IR-RF database entry	184
EmberAfRf4ceMsoIrRfDatabaseIrDescriptor	
RF4CE MSO IR-RF database IR descriptor	185

EmberAfRf4ceMsoIrRfDescriptor	
RF4CE MSO IR-RF database RF descriptor	186
EmberAfRf4ceMsoUserControlRecord	
This data structure contains the MSO user control record	187
EmberAfRf4ceZrcActionMapping	
RF4CE ZRC Action Mapping	188
EmberAfRf4ceZrcActionRecord	
This data structure contains the ZRC action record	189
EmberAfRf4ceZrcCommandsSupported	
This data structure contains the ZRC 1.x command discovery data	191
EmberAfRf4ceZrcHomeAutomationAttribute	
RF4CE ZRC Home Automation attribute	191
EmberAfRf4ceZrcHomeAutomationSupported	
RF4CE ZRC Home Automation supported	192
EmberAfRf4ceZrcMappableAction	
RF4CE ZRC Mappable Action	193
EmberAfRf4ceZrcUserControlRecord	
This data structure contains the ZRC 1.x user control record	193
HaAttributesInfo	194

Chapter 5

File Index

5.1 File List

Here is a list of all files with brief descriptions:

_AF_RF4CE.top	196
lv-shutdown.h	202
rf4ce-gdp-attributes.h	212
rf4ce-gdp-identification-client.h	216
rf4ce-gdp-identification-server.h	216
rf4ce-gdp-identification.h	218
rf4ce-gdp-internal.h	246
rf4ce-gdp-poll.h	258
rf4ce-gdp-test.h	263
rf4ce-gdp-tokens.h	266
rf4ce-gdp-types.h	274
rf4ce-gdp.h	280
rf4ce-mso-attributes.h	286
rf4ce-mso-internal.h	301
rf4ce-mso-ir-rf-database-originator.h	306
rf4ce-mso-ir-rf-database-recipient.h	307
rf4ce-mso-test.h	309
rf4ce-mso-tokens.h	310
rf4ce-mso-types.h	323
rf4ce-mso.h	331
rf4ce-profile-internal.h	337
rf4ce-profile-types.h	342
rf4ce-profile.h	345
rf4ce-zrc11-internal.h	353
rf4ce-zrc11-types.h	360
rf4ce-zrc11.h	363
rf4ce-zrc20-action-mapping-client.h	364
rf4ce-zrc20-action-mapping-server.h	366
rf4ce-zrc20-action-mapping.h	370
rf4ce-zrc20-attributes.h	376
rf4ce-zrc20-ha-actions.h	380
rf4ce-zrc20-ha-client.h	384
rf4ce-zrc20-ha-server-tokens.h	392

rf4ce-zrc20-ha-server.h	396
rf4ce-zrc20-internal.h	413
rf4ce-zrc20-test.h	421
rf4ce-zrc20-tokens.h	424
rf4ce-zrc20-types.h	440
rf4ce-zrc20.h	450

Chapter 6

Module Documentation

6.1 Ember ZigBee RF4CE API Reference

Modules

- [Generic Device Profile](#)
- [Generic Device Profile Identification Client](#)
- [Generic Device Profile Identification Server](#)
- [RF4CE Multiple System Operators Profile](#)
- [RF4CE Multiple System Operators IR-RF Database Originator](#)
- [RF4CE Multiple System Operators IR-RF Database Recipient](#)
- [RF4CE Profile Support](#)
- [ZigBee Remote Control 1.1 Profile](#)
- [ZigBee Remote Control 2.0 Profile](#)
- [ZigBee Remote Control 2.0 Action Mapping Client](#)
- [ZigBee Remote Control 2.0 Action Mapping Server](#)
- [ZigBee Remote Control 2.0 Home Automation Client](#)
- [ZigBee Remote Control 2.0 Home Automation Server](#)

6.1.1 Detailed Description

The Ember ZigBee RF4CE API Reference includes the documentation for writing applications using the RF4CE plugins included in this framework. A customer is also encouraged to view the callback interface documentation for these plugins, which can be found at [Ember RF4CE Application Framework Callbacks](#).

6.2 Generic Device Profile

Macros

- #define EMBER_AF_PLUGIN_RF4CE_GDP_IS_ORIGINATOR
- #define POLL_CLIENT
- #define POLL_SERVER
- #define IDENTIFICATION_CLIENT
- #define IDENTIFICATION_SERVER

Functions

- uint8_t * **emberAfRf4ceGdpRandContents** (EmberAfRf4ceGdpRand *rand)
- uint8_t * **emberAfRf4ceGdpTagContents** (EmberAfRf4ceGdpTag *tag)
- EmberStatus **emberAfRf4ceGdpBind** (uint8_t *profileIdList, uint8_t profileIdListLength, uint8_t searchDevType)
- EmberStatus **emberAfRf4ceGdpProxyBind** (EmberPanId panId, EmberEUI64 ieeeAddr, uint8_t *profileIdList, uint8_t profileIdListLength)
- void **emberAfRf4ceGdpConfigurationProcedureComplete** (bool success)
- void **emberAfRf4ceGdpPushButton** (bool setPending)
- void **emberAfRf4ceGdpSetValidationStatus** (EmberAfRf4ceGdpCheckValidationStatus status)
- EmberStatus **emberAfRf4ceGdpInitiateKeyExchange** (uint8_t pairingIndex)
- EmberStatus **emberAfRf4ceGdpPoll** (uint8_t pairingIndex, uint16_t vendorId, **EmberAfRf4ceGdpHeartbeatTrigger** trigger)
- bool **emberAfRf4ceGdpMessageSent** (uint8_t pairingIndex, uint8_t profileId, uint16_t vendorId, const uint8_t *message, uint8_t messageLength, EmberStatus status)
- bool **emberAfRf4ceGdpIncomingMessage** (uint8_t pairingIndex, uint8_t profileId, uint16_t vendorId, EmberRf4ceTxOption secured, const uint8_t *message, uint8_t messageLength)
- EmberStatus **emberAfRf4ceGdpGetCommandTxOptions** (EmberAfRf4ceGdpCommandCode commandCode, uint8_t pairingIndex, uint16_t vendorId, EmberRf4ceTxOption *txOptions)
- EmberStatus **emberAfRf4ceGdpGenericResponse** (uint8_t pairingIndex, uint8_t profileId, uint16_t vendorId, **EmberAfRf4ceGdpResponseCode** responseCode)
- EmberStatus **emberAfRf4ceGdpConfigurationComplete** (uint8_t pairingIndex, uint8_t profileId, uint16_t vendorId, **EmberAfRf4ceGdpStatus** status)
- EmberStatus **emberAfRf4ceGdpHeartbeat** (uint8_t pairingIndex, uint16_t vendorId, **EmberAfRf4ceGdpHeartbeatTrigger** trigger)
- EmberStatus **emberAfRf4ceGdpGetAttributes** (uint8_t pairingIndex, uint8_t profileId, uint16_t vendorId, const **EmberAfRf4ceGdpAttributeIdentificationRecord** *records, uint8_t recordsLength)
- EmberStatus **emberAfRf4ceGdpGetAttributesResponse** (uint8_t pairingIndex, uint8_t profileId, uint16_t vendorId, const **EmberAfRf4ceGdpAttributeStatusRecord** *records, uint8_t recordsLength)
- EmberStatus **emberAfRf4ceGdpPushAttributes** (uint8_t pairingIndex, uint8_t profileId, uint16_t vendorId, const **EmberAfRf4ceGdpAttributeRecord** *records, uint8_t recordsLength)
- EmberStatus **emberAfRf4ceGdpSetAttributes** (uint8_t pairingIndex, uint8_t profileId, uint16_t vendorId, const **EmberAfRf4ceGdpAttributeRecord** *records, uint8_t recordsLength)
- EmberStatus **emberAfRf4ceGdpPullAttributes** (uint8_t pairingIndex, uint8_t profileId, uint16_t vendorId, const **EmberAfRf4ceGdpAttributeIdentificationRecord** *records, uint8_t recordsLength)
- EmberStatus **emberAfRf4ceGdpPullAttributesResponse** (uint8_t pairingIndex, uint8_t profileId, uint16_t vendorId, const **EmberAfRf4ceGdpAttributeStatusRecord** *records, uint8_t recordsLength)
- EmberStatus **emberAfRf4ceGdpCheckValidationRequest** (uint8_t pairingIndex, uint16_t vendorId, uint8_t control)

- EmberStatus `emberAfRf4ceGdpCheckValidationResponse` (uint8_t pairingIndex, uint16_t vendorId, `EmberAfRf4ceGdpCheckValidationStatus` status)
- EmberStatus `emberAfRf4ceGdpClientNotificationIdentify` (uint8_t pairingIndex, uint16_t vendorId, `EmberAfRf4ceGdpClientNotificationIdentifyFlags` flags, uint16_t timeS)
- EmberStatus `emberAfRf4ceGdpClientNotificationRequestPollNegotiation` (uint8_t pairingIndex, uint16_t vendorId)
- EmberStatus `emberAfRf4ceGdpClientNotification` (uint8_t pairingIndex, uint8_t profileId, uint16_t vendorId, uint8_t subtype, const uint8_t *payload, uint8_t payloadLength)
- EmberStatus `emberAfRf4ceGdpSubscribeToHeartbeat` (`EmberAfRf4ceGdpHeartbeatCallback` callback)

6.2.1 Detailed Description

The Generic Device Profile (GDP) plugin provides APIs to service the different RF4CE profiles.

The plugin offers APIs to add in application level customization to the RF4CE GDP commissioning mechanism. One can initiate this commissioning process with `emberAfRf4ceGdpBind` or `emberAfRf4ceGdpProxyBind`. The application can use `emberAfRf4ceGdpConfigurationComplete` to learn the result of this binding operation.

One can configure basic security operations of an RF4CE network using this plugin. The plugin offers options Enhanced security, Standard shared secret, and Vendor-specific shared secrets, so that the user can dictate the level of security at which they would like this plugin to operate. During the commissioning process, or any time after, the application can use `emberAfRf4ceGdpInitiateKeyExchange` to force a key exchange to take place.

This plugin also gives the application the ability to interact with RF4CE attributes. There are APIs provided to perform the necessary GDP attribute operations over the air.

6.2.2 Macro Definition Documentation

6.2.2.1 #define EMBER_AF_PLUGIN_RF4CE_GDP_IS_ORIGINATOR

Set if the RF4CE Profile plugin was configured as a controller or target.

Definition at line 40 of file `rf4ce-gdp.h`.

6.2.2.2 #define POLL_CLIENT

Definition at line 46 of file `rf4ce-gdp.h`.

6.2.2.3 #define POLL_SERVER

Definition at line 47 of file `rf4ce-gdp.h`.

6.2.2.4 #define IDENTIFICATION_CLIENT

Definition at line 57 of file `rf4ce-gdp.h`.

6.2.2.5 #define IDENTIFICATION_SERVER

Definition at line 58 of file [rf4ce-gdp.h](#).

6.2.3 Function Documentation

6.2.3.1 uint8_t* emberAfRf4ceGdpRandContents (EmberAfRf4ceGdpRand * rand)

Accesses the actual random byte string of the [EmberAfRf4ceGdpRand](#) structure.

Parameters

<i>rand</i>	A pointer to an EmberAfRf4ceGdpRand structure.
-------------	--

Returns

uint8_t* Returns a pointer to the first byte of the random byte string.

6.2.3.2 uint8_t* emberAfRf4ceGdpTagContents (EmberAfRf4ceGdpTag * tag)

Accesses the actual tag value of the [EmberAfRf4ceGdpTag](#) structure.

Parameters

<i>tag</i>	A pointer to an EmberAfRf4ceGdpTag structure.
------------	---

Returns

uint8_t* Returns a pointer to the first byte of the tag value.

6.2.3.3 EmberStatus emberAfRf4ceGdpBind (uint8_t * profileIdList, uint8_t profileIdListLength, uint8_t searchDevType)

Initiates the binding process.

Parameters

<i>profileIdList</i>	The list of profile IDs supported by the node.
<i>profileIdListLength</i>	The size of the profile ID list.
<i>searchDevType</i>	The device type the node will be matching against during the preliminary discovery process.

Returns

An ::EmberStatus value indicating whether the binding process was successfully initiated or the reason of failure.

6.2.3.4 EmberStatus emberAfRf4ceGdpProxyBind (EmberPanId *panId*, EmberEUI64 *ieeeAddr*, uint8_t * *profileIdList*, uint8_t *profileIdListLength*)

Initiates the proxy binding process.

Parameters

<i>panId</i>	The pan ID of the recipient.
<i>ieeeAddr</i>	The IEEE address of the recipient.
<i>profileIdList</i>	The list of profile IDs supported by the node.
<i>profileIdListLength</i>	The size of the profile ID list.

Returns

An ::EmberStatus value indicating whether the proxy binding process was successfully initiated or the reason of failure.

6.2.3.5 void emberAfRf4ceGdpConfigurationProcedureComplete (bool *success*)

It notifies the GDP profile that a profile-specific configuration procedure has completed.

Parameters

<i>success</i>	Indicates whether the profile-specific configuration procedure completed successfully or not.
----------------	---

6.2.3.6 void emberAfRf4ceGdpPushButton (bool *setPending*)

It sets or clears the push button stimulus pending flag at the GDP recipient.

Parameters

<i>The</i>	push button stimulus pending flag.
------------	------------------------------------

6.2.3.7 void emberAfRf4ceGdpSetValidationStatus (EmberAfRf4ceGdpCheckValidationStatus *status*)

It sets the validation status at the GDP recipient.

Parameters

<i>status</i>	The validation status.
---------------	------------------------

6.2.3.8 EmberStatus emberAfRf4ceGdpInitiateKeyExchange (uint8_t *pairingIndex*)

It kicks off the extended key exchange procedure at the GDP originator or recipient.

Parameters

<i>The</i>	pairing index for which the key exchange procedure should be executed.
------------	--

6.2.3.9 EmberStatus emberAfRf4ceGdpPoll (*uint8_t pairingIndex, uint16_t vendorId, EmberAfRf4ceGdpHeartbeatTrigger trigger*)

It polls the poll server by sending out a GDP heartbeat command to the poll server.

Parameters

<i>pairingIndex</i>	The pairing index of the poll server.
<i>vendorId</i>	The vendor ID to be included in the heartbeat command.
<i>trigger</i>	The heartbeat trigger to be included in the heartbeat command.

Returns

Indicates whether the heartbeat command was successfully sent out or the reason of failure.

6.2.3.10 bool emberAfRf4ceGdpMessageSent (*uint8_t pairingIndex, uint8_t profileId, uint16_t vendorId, const uint8_t * message, uint8_t messageLength, EmberStatus status*)

Submit a message sent indication to the GDP plugin for any applicable GDP-specific processing.

The application should submit all message sent indications for profiles that use GDP to this plugin before performing any profile-specific processing. Profile-specific processing should only occur if this API returns false.

Returns

true if the message sent indication was for a GDP command and has been handled by the GDP plugin and should therefore not be processed further by the application or false if the message sent indication was not for a GDP command and the application should perform profile-specific processing.

6.2.3.11 bool emberAfRf4ceGdpIncomingMessage (*uint8_t pairingIndex, uint8_t profileId, uint16_t vendorId, EmberRf4ceTxOption secured, const uint8_t * message, uint8_t messageLength*)

Submit an incoming message to the GDP plugin for any applicable GDP- specific processing.

The application should submit all incoming messages for profiles that use GDP to this plugin before performing any profile-specific processing. Profile-specific processing should only occur if this API returns false.

Returns

true if the incoming message was a GDP command and has been handled by the GDP plugin and should therefore not be processed further by the application or false if the incoming message was not a GDP command and the application should perform profile-specific processing.

6.2.3.12 EmberStatus emberAfRf4ceGdpGetCommandTxOptions (EmberAfRf4ceGdpCommandCode *commandCode*, uint8_t *pairingIndex*, uint16_t *vendorId*, EmberRf4ceTxOption * *txOptions*)

Allows the application to retrieve the RF4CE TX options used by GDP for a specific command, pairing index and vendor ID.

Returns

An ::EmberStatus indicating success or the reason of failure.

6.2.3.13 EmberStatus emberAfRf4ceGdpGenericResponse (uint8_t *pairingIndex*, uint8_t *profileId*, uint16_t *vendorId*, EmberAfRf4ceGdpResponseCode *responseCode*)

It sends out a GDP generic response command.

Parameters

<i>paringIndex</i>	The pairing index the generic response command should be sent to.
<i>profileId</i>	The profile ID to be included in the generic response command.
<i>vendorId</i>	The vendor ID to be included in the generic response command.
<i>responseCode</i>	The response code to be included in the generic response command.

Returns

An ::EmberStatus value indicating whether the generic response command was successfully sent out or the reason of failure.

6.2.3.14 EmberStatus emberAfRf4ceGdpConfigurationComplete (uint8_t *pairingIndex*, uint8_t *profileId*, uint16_t *vendorId*, EmberAfRf4ceGdpStatus *status*)

It sends out a GDP configuration complete command.

Parameters

<i>paringIndex</i>	The pairing index the configuration complete command should be sent to.
<i>profileId</i>	The profile ID to be included in the configuration complete command.
<i>vendorId</i>	The vendor ID to be included in the configuration complete command.
<i>status</i>	The status code to be included in the configuration complete command.

Returns

An ::EmberStatus value indicating whether the configuration complete command was successfully sent out or the reason of failure.

6.2.3.15 EmberStatus emberAfRf4ceGdpHeartbeat (uint8_t *pairingIndex*, uint16_t *vendorId*, EmberAfRf4ceGdpHeartbeatTrigger *trigger*)

It sends out a GDP hearbeat command.

Parameters

<i>paringIndex</i>	The pairing index the heartbeat command should be sent to.
<i>vendorId</i>	The vendor ID to be included in the heartbeat command.
<i>trigger</i>	The trigger code to be included in the heartbeat command.

Returns

An ::EmberStatus value indicating whether the heartbeat command was successfully sent out or the reason of failure.

6.2.3.16 EmberStatus emberAfRf4ceGdpGetAttributes (uint8_t *pairingIndex*, uint8_t *profileId*, uint16_t *vendorId*, const EmberAfRf4ceGdpAttributeIdentificationRecord * *records*, uint8_t *recordsLength*)

It sends out a GDP get attributes command.

Parameters

<i>paringIndex</i>	The pairing index the get attributes command should be sent to.
<i>profileId</i>	The profile ID to be included in the get attributes command.
<i>vendorId</i>	The vendor ID to be included in the get attributes command.
<i>records</i>	A list of EmberAfRf4ceGdpAttributeIdentificationRecord to be included in the get attributes command.
<i>recordsLength</i>	The size of the attribute identification records list.

Returns

An ::EmberStatus value indicating whether the get attributes command was successfully sent out or the reason of failure.

6.2.3.17 EmberStatus emberAfRf4ceGdpGetAttributesResponse (uint8_t *pairingIndex*, uint8_t *profileId*, uint16_t *vendorId*, const EmberAfRf4ceGdpAttributeStatusRecord * *records*, uint8_t *recordsLength*)

It sends out a GDP get attributes response command.

Parameters

<i>paringIndex</i>	The pairing index the get attributes response command should be sent to.
<i>profileId</i>	The profile ID to be included in the get attributes response command.
<i>vendorId</i>	The vendor ID to be included in the get attributes response command.
<i>records</i>	A list of EmberAfRf4ceGdpAttributeStatusRecord to be included in the get attributes response command.
<i>recordsLength</i>	The size of the attribute status records list.

Returns

An ::EmberStatus value indicating whether the get attributes response command was successfully sent out or the reason of failure.

6.2.3.18 EmberStatus emberAfRf4ceGdpPushAttributes (*uint8_t pairingIndex, uint8_t profileId, uint16_t vendorId, const EmberAfRf4ceGdpAttributeRecord * records, uint8_t recordsLength*)

It sends out a GDP push attributes command.

Parameters

<i>paringIndex</i>	The pairing index the push attributes command should be sent to.
<i>profileId</i>	The profile ID to be included in the push attributes command.
<i>vendorId</i>	The vendor ID to be included in the push attributes command.
<i>records</i>	A list of EmberAfRf4ceGdpAttributeRecord to be included in the push attributes command.
<i>recordsLength</i>	The size of the attribute records list.

Returns

An `:EmberStatus` value indicating whether the push attributes command was successfully sent out or the reason of failure.

6.2.3.19 EmberStatus emberAfRf4ceGdpSetAttributes (*uint8_t pairingIndex, uint8_t profileId, uint16_t vendorId, const EmberAfRf4ceGdpAttributeRecord * records, uint8_t recordsLength*)

It sends out a GDP set attributes command.

Parameters

<i>paringIndex</i>	The pairing index the set attributes command should be sent to.
<i>profileId</i>	The profile ID to be included in the set attributes command.
<i>vendorId</i>	The vendor ID to be included in the set attributes command.
<i>records</i>	A list of EmberAfRf4ceGdpAttributeRecord to be included in the set attributes command.
<i>recordsLength</i>	The size of the attribute records list.

Returns

An `:EmberStatus` value indicating whether the set attributes command was successfully sent out or the reason of failure.

6.2.3.20 EmberStatus emberAfRf4ceGdpPullAttributes (*uint8_t pairingIndex, uint8_t profileId, uint16_t vendorId, const EmberAfRf4ceGdpAttributeIdentificationRecord * records, uint8_t recordsLength*)

It sends out a GDP pull attributes command.

Parameters

<i>paringIndex</i>	The pairing index the pull attributes command should be sent to.
<i>profileId</i>	The profile ID to be included in the pull attributes command.
<i>vendorId</i>	The vendor ID to be included in the pull attributes command.
<i>records</i>	A list of EmberAfRf4ceGdpAttributeIdentificationRecord to be included in the pull attributes command.
<i>recordsLength</i>	The size of the attribute identification records list.

Returns

An ::EmberStatus value indicating whether the pull attributes command was successfully sent out or the reason of failure.

6.2.3.21 EmberStatus emberAfRf4ceGdpPullAttributesResponse (uint8_t pairingIndex, uint8_t profileId, uint16_t vendorId, const EmberAfRf4ceGdpAttributeStatusRecord * records, uint8_t recordsLength)

It sends out a GDP pull attributes response command.

Parameters

<i>paringIndex</i>	The pairing index the pull attributes response command should be sent to.
<i>profileId</i>	The profile ID to be included in the pull attributes response command.
<i>vendorId</i>	The vendor ID to be included in the pull attributes response command.
<i>records</i>	A list of EmberAfRf4ceGdpAttributeStatusRecord to be included in the pull attributes response command.
<i>recordsLength</i>	The size of the attribute status records list.

Returns

An ::EmberStatus value indicating whether the pull attributes response command was successfully sent out or the reason of failure.

6.2.3.22 EmberStatus emberAfRf4ceGdpCheckValidationRequest (uint8_t pairingIndex, uint16_t vendorId, uint8_t control)

It sends out a GDP check validation request command.

Parameters

<i>paringIndex</i>	The pairing index the check validation request command should be sent to.
<i>vendorId</i>	The vendor ID to be included in the check validation request command.
<i>control</i>	The control field to be included in the check validation request command.

Returns

An ::EmberStatus value indicating whether the check validation request command was successfully sent out or the reason of failure.

6.2.3.23 EmberStatus emberAfRf4ceGdpCheckValidationResponse (uint8_t pairingIndex, uint16_t vendorId, EmberAfRf4ceGdpCheckValidationStatus status)

It sends out a GDP check validation response command.

Parameters

<i>paringIndex</i>	The pairing index the check validation response command should be sent to.
<i>vendorId</i>	The vendor ID to be included in the check validation response command.
<i>status</i>	The status code to be included in the check validation response command.

Returns

An ::EmberStatus value indicating whether the check validation response command was successfully sent out or the reason of failure.

6.2.3.24 EmberStatus emberAfRf4ceGdpClientNotificationIdentify (uint8_t pairingIndex, uint16_t vendorId, EmberAfRf4ceGdpClientNotificationIdentifyFlags flags, uint16_t timeS)

It sends out a GDP client notification identify command.

Parameters

<i>paringIndex</i>	The pairing index the client notification identify command should be sent to.
<i>vendorId</i>	The vendor ID to be included in the client notification identify command.
<i>flags</i>	The flags field to be included in the client notification identify command.
<i>timeS</i>	The time field in seconds to be included in the client notification identify command.

Returns

An ::EmberStatus value indicating whether the client notification identify command was successfully sent out or the reason of failure.

6.2.3.25 EmberStatus emberAfRf4ceGdpClientNotificationRequestPollNegotiation (uint8_t pairingIndex, uint16_t vendorId)

It sends out a GDP client notification request poll negotiation command.

Parameters

<i>paringIndex</i>	The pairing index the client notification request poll negotiation command should be sent to.
<i>vendorId</i>	The vendor ID to be included in the client notification request poll negotiation command.

Returns

An ::EmberStatus value indicating whether the client notification request poll negotiation command was successfully sent out or the reason of failure.

6.2.3.26 EmberStatus emberAfRf4ceGdpClientNotification (uint8_t pairingIndex, uint8_t profileId, uint16_t vendorId, uint8_t subtype, const uint8_t * payload, uint8_t payloadLength)

It sends out a generic GDP client notification command.

Parameters

<i>paringIndex</i>	The pairing index the client notification command should be sent to.
<i>vendorId</i>	The vendor ID to be included in the client notification command.
<i>subType</i>	The sub-type field of the client notification command.
<i>payload</i>	The payload to be included in the client notification command.
<i>payloadLength</i>	The length in bytes of the payload.

Returns

An ::EmberStatus value indicating whether the client notification command was successfully sent out or the reason of failure.

6.2.3.27 EmberStatus emberAfRf4ceGdpSubscribeToHeartbeat(EmberAfRf4ceGdpHeartbeatCallback callback)

It allows a software module to subscribe to incoming heartbeat commands.

Parameters

<i>callback</i>	The callback to be called by the poll server upon receiving a heartbeat command.
-----------------	--

Returns

An ::EmberStatus value of ::EMBER_SUCCESS if the subscription process succeeded. An ::EmberStatus value of ::EMBER_TABLE_FULL if there is no room in the subscription table. An ::EmberStatus of ::EMBER_INVALID_CALL if the node is not a poll server.

6.3 Generic Device Profile Identification Client

Functions

- void [emberAfRf4ceGdpIdentificationClientDetectedUserInteraction](#) (void)

6.3.1 Detailed Description

This plugin provides an implementation of the Generic Device Profile (GDP) Identification Client. It offers an excellent example of the role of an identification client under GDP using the Silicon Labs development kit. However, if a customer moves their application to different hardware, they will need to create their own implementation that makes use of the new hardware. See UG10310 for more information regarding the role of the GDP Identification Client.

In this Ember implementation of the GDP Identification Client, indicators on the Silicon Labs development board will react to identification commands from the server, notifying the user that it is in identifying mode. These notifications come in the form of flashing LEDs, a buzzer sounding, or printing status messages.

The application can notify the plugin that a user interaction has been detected using the API [emberAfRf4ce-GdpIdentificationClientDetectedUserInteraction](#). This will trigger the plugin to perform the necessary actions in accordance with the GDP standard. A user can also incite call upon this event by issuing the following command over the command line interface: *plugin rf4ce-gdp-identification-client user-interaction*.

Please see [Generic Device Profile Identification Server](#) for an implementation of the RF4CE GDP Identification Server.

6.3.2 Function Documentation

6.3.2.1 void [emberAfRf4ceGdpIdentificationClientDetectedUserInteraction](#) (void)

Notify the plugin that a user interaction has been detected. Note that this command can also be called from the command line interface using the command *plugin rf4ce-gdp-identification-client user-interaction*.

6.4 Generic Device Profile Identification Server

Functions

- EmberStatus [emberAfRf4ceGdpIdentificationServerIdentify](#) (uint8_t pairingIndex, [EmberAfRf4ceGdpClientNotificationIdentifyFlags](#) flags, uint16_t timeS)

6.4.1 Detailed Description

This plugin provides an implementation of the RF4CE Generic Device Profile (GDP) Identification Server. It gives the user the opportunity to manage the GDP identification process from the server side.

The application can command the server to send an identify command to a client using the API [emberAfRf4ceGdpIdentificationServerIdentify](#). One can also begin the process over the command line interface by using the command *plugin rf4ce-gdp-identification-server identify* and supplying the desired arguments.

Please see [rf4ce-gdp-identification-Client](#) for an implementation of the RF4CE GDP Identification Client. Unlike the RF4CE GDP Identification Client plugin, this plugin is able to be used across hardware platforms.

6.4.2 Function Documentation

6.4.2.1 EmberStatus [emberAfRf4ceGdpIdentificationServerIdentify](#) (uint8_t *pairingIndex*, [EmberAfRf4ceGdpClientNotificationIdentifyFlags](#) *flags*, uint16_t *timeS*)

Queue up an identify command to be sent to an identification client in the pairing table. Note that one can issue the command *plugin rf4ce-gdp-identification-server identify* over the command line interface in order to perform the same action.

Parameters

<i>pairingIndex</i>	The index in the pairing table to which to send the identification command.
<i>flags</i>	The flags to send in this command. See data type EmberAfRf4ceGdpClientNotificationIdentifyFlags for more information on this parameter.
<i>timeS</i>	The time in seconds for which the client should identify.

Returns

An ::EmberStatus value indicating the success of the attempt to queue up the identify command.

6.5 RF4CE Multiple System Operators Profile

Macros

- #define EMBER_AF_PLUGIN_RF4CE_MSO_IS_RECIPIENT

Functions

- EmberStatus `emberAfRf4ceMsoBind` (void)
- EmberStatus `emberAfRf4ceMsoWatchdogKick` (uint16_t validationWatchdogTimeMs)
- EmberStatus `emberAfRf4ceMsoValidate` (void)
- EmberStatus `emberAfRf4ceMsoTerminateValidation` (void)
- EmberStatus `emberAfRf4ceMsoAbortValidation` (bool fullAbort)
- EmberStatus `emberAfRf4ceMsoUserControlPress` (uint8_t pairingIndex, `EmberAfRf4ceMsoKeyCode` rcCommandCode, const uint8_t *rcCommandPayload, uint8_t rcCommandPayloadLength, bool atomic)
- EmberStatus `emberAfRf4ceMsoUserControlRelease` (uint8_t pairingIndex, `EmberAfRf4ceMsoKeyCode` rcCommandCode)
- EmberStatus `emberAfRf4ceMsoSetAttributeRequest` (uint8_t pairingIndex, `EmberAfRf4ceMsoAttributeId` attributeId, uint8_t index, uint8_t valueLen, const uint8_t *value)
- EmberStatus `emberAfRf4ceMsoGetAttributeRequest` (uint8_t pairingIndex, `EmberAfRf4ceMsoAttributeId` attributeId, uint8_t index, uint8_t valueLen)
- bool `emberAfRf4ceMsoIrRfDatabaseEntryUseDefault` (const `EmberAfRf4ceMsoIrRfDatabaseEntry` *entry)
- bool `emberAfRf4ceMsoIrRfDatabaseEntryHasRfPressedDescriptor` (const `EmberAfRf4ceMsoIrRfDatabaseEntry` *entry)
- bool `emberAfRf4ceMsoIrRfDatabaseEntryHasRfRepeatedDescriptor` (const `EmberAfRf4ceMsoIrRfDatabaseEntry` *entry)
- bool `emberAfRf4ceMsoIrRfDatabaseEntryHasRfReleasedDescriptor` (const `EmberAfRf4ceMsoIrRfDatabaseEntry` *entry)
- uint8_t `emberAfRf4ceMsoIrRfDatabaseEntryGetMinimumNumberOfTransmissions` (const `EmberAfRf4ceMsoIrRfDescriptor` *rfDescriptor)
- bool `emberAfRf4ceMsoIrRfDatabaseEntryShouldTransmitUntilRelease` (const `EmberAfRf4ceMsoIrRfDescriptor` *rfDescriptor)
- bool `emberAfRf4ceMsoIrRfDatabaseEntryHasIrDescriptor` (const `EmberAfRf4ceMsoIrRfDatabaseEntry` *entry)

6.5.1 Detailed Description

The RF4CE Multiple System Operators (MSO) plugin implements the MSO profile. The profile uses the simple user control procedure from the ZigBee Remote Control (ZRC) profile with a more robust discovery, pairing, and validating procedure for binding devices. The plugin manages these procedures for both originators and recipients.

This plugin supports originators and recipients. Originators send action messages to recipients after binding with them. Unlike ZRC, originators must be controllers. Recipients must be targets, as in ZRC. Both controllers and targets must start general network operations before beginning MSO-specific operations. Network operations should be started by calling `emberAfRf4ceStart` in the `RF4CE Profile Support` plugin.

Once network operations have started, controllers can initiate the binding procedure by calling [emberAfRf4ceMsoBind](#). The plugin will perform discovery for matching targets in range. Potential targets are ranked according to an algorithm described in the MSO specification. If one or more potential targets are identified, the plugin will attempt to pair with the highest-ranked target. Once the temporary pairing is established, the validation procedure begins. Validation is implementation-specific, but may be as simple as a button press on the target or a more involved challenge-response mechanism between target and originator. Only if validation is successful are the controller and target considered bound. If any step fails, the plugin will restart the binding procedure using the target that has the next-highest rank. At the conclusion of the binding procedure, the plugin will call [emberAfPluginRf4ceMsoBindingCompleteCallback](#) to indicate whether binding completed successfully with a target and, if so, which pairing index has been assigned to that target.

The plugin manages discovery and pairing for targets on behalf of the device. If pairing completes successfully, the plugin will call [emberAfPluginRf4ceMsoStartValidationCallback](#) so that the device can begin the implementation-specific validation procedure. During validation, the application must periodically call [emberAfRf4ceMsoWatchdogKick](#) to prevent a validation timeout. Once the application determines the controller has validated successfully, it should call [emberAfRf4ceMsoValidate](#). If the controller fails validation, the application may instead call [emberAfRf4ceMsoTerminateValidation](#) or [emberAfRf4ceMsoAbortValidation](#). Once binding completes, the plugin will call [emberAfPluginRf4ceMsoBindingCompleteCallback](#) to indicate whether binding completed successfully with a controller and, if so, which pairing index has been assigned to that controller.

Following a successful discovery and pairing, controllers may send user control messages to targets by calling [emberAfRf4ceMsoUserControlPress](#). In response, the plugin will transmit a user control press message to the indicated target with the HDMI-CEC command code and payload. If the press is atomic, the plugin performs no additional processing. Otherwise, the plugin will repeatedly transmit user control repeat messages at fixed intervals until [emberAfRf4ceMsoUserControlRelease](#) is called. The interval at which user control repeat messages are transmitted is configurable in the plugin options.

For targets, the plugin will keep track of incoming user control messages. Each time a user control press command is received, the plugin will call [emberAfPluginRf4ceMsoUserControlCallback](#) with its HDMI-CEC command code and payload. The plugin will then wait for a fixed duration for a corresponding user control repeat or release messages. If a user control repeat message is received within the timeout, the plugin will reset its timer and wait for the next message. If a user control repeat message is not received within the timeout or if a user control release is received, the plugin will call [emberAfPluginRf4ceMsoUserControlCallback](#) with an indication that the user control has stopped. The timeout for receiving repeat messages is configurable in the plugin options. Note that the plugin will not call [emberAfPluginRf4ceMsoUserControlCallback](#) for repeat messages that follow a press message.

The plugin is capable of keeping track of a fixed number of simultaneous incoming and outgoing user control messages. The limits are configurable in the plugin options.

This plugin manages the state of the receiver by calling [emberAfRf4ceRxEnable](#) using [EMBER_AF_RF4CE_PROFILE_MSO](#) as the profile id. If the application also wishes to manage the receiver, it should do so using [EMBER_AF_RF4CE_PROFILE_WILDCARD](#) as the profile id or by calling [emberAfRf4ceSetPowerSavingParameters](#).

This plugin utilizes the discovery, pairing, sending and receiving, and power-saving functionality provided by the [RF4CE Profile Support](#) plugin. Support for the optional IR-RF database feature is provided by the [RF4CE Multiple System Operators IR-RF Database Originator](#) and [RF4CE Multiple System Operators IR-RF Database Recipient](#) plugins.

6.5.2 Macro Definition Documentation

6.5.2.1 `#define EMBER_AF_PLUGIN_RF4CE_MSO_IS_RECIPIENT`

Set if the RF4CE Profile plugin was configured as a target.

Definition at line 103 of file [rf4ce-mso.h](#).

6.5.3 Function Documentation

6.5.3.1 `EmberStatus emberAfRf4ceMsoBind (void)`

Initiate the binding procedure.

The plugin begins the binding procedure by searching for and ranking potential targets with which to pair. If one or more targets is identified, the plugin will create a temporary pairing with the highest-ranked target. Once the temporary pairing completes, the application should perform the required validation procedure. During this time, the plugin will periodically query the target for the validation status. If validation completes successfully, the plugin will notify the application by calling [emberAfPluginRf4ceMsoBindingCompleteCallback](#). If the temporary pairing fails or if validation fails, the plugin will attempt to bind to the target with the next-highest rank. If the plugin fails to bind with any target, it will call [emberAfPluginRf4ceMsoBindingCompleteCallback](#) with an error.

This function is only applicable to target devices.

Returns

An `:EmberStatus` value that indicates the success or failure of the command.

6.5.3.2 `EmberStatus emberAfRf4ceMsoWatchdogKick (uint16_t validationWatchdogTimeMs)`

Kick the watchdog.

If the plugin was configured with a non-zero initial watchdog time, this function must be called during validation to prevent the watchdog timer from expiring and triggering a TIMEOUT failure. The first call should occur within the initial watchdog time of the call to [emberAfPluginRf4ceMsoStartValidationCallback](#). To avoid further TIMEOUTs, each call to this function must either be followed either by a subsequent call within the provided timeout or by a call to [emberAfRf4ceMsoValidate](#), [emberAfRf4ceMsoTerminateValidation](#), or [emberAfRf4ceMsoAbortValidation](#).

This function is only applicable to target devices.

Parameters

<code>validation-WatchdogTime-Ms</code>	The time in milliseconds to reset the watchdog timer or zero to disable the watchdog timer.
---	---

Returns

An `:EmberStatus` value that indicates the success or failure of the command.

6.5.3.3 `EmberStatus emberAfRf4ceMsoValidate (void)`

Validate a controller.

This function can be called to indicate the validation procedure completed successfully. It should be called if the controller has performed the required validation procedure satisfactorily.

This function is only applicable to target devices.

Returns

An ::EmberStatus value that indicates the success or failure of the command.

6.5.3.4 EmberStatus emberAfRf4ceMsoTerminateValidation (void)

Terminate the validation procedure.

This function can be called to terminate the validation procedure. It should be called if the controller fails validation.

This function is only applicable to target devices.

Returns

An ::EmberStatus value that indicates the success or failure of the command.

6.5.3.5 EmberStatus emberAfRf4ceMsoAbortValidation (bool fullAbort)

Abort the validation procedure.

This function can be called to abort the validation procedure. It should be called in response to a controller sending the Abort or FullAbort keys during validation.

Parameters

<i>fullAbort</i>	true if the controller should not attempt to validate with other controllers or false if the controller should attempt to validate with other controller.
------------------	---

This function is only applicable to target devices.

Returns

An ::EmberStatus value that indicates the success or failure of the command.

6.5.3.6 EmberStatus emberAfRf4ceMsoUserControlPress (uint8_t pairingIndex, EmberAfRf4ceMsoKeyCode rcCommandCode, const uint8_t * rcCommandPayload, uint8_t rcCommandPayloadLength, bool atomic)

Send indications of user control presses and repeats to a pairing index.

This function can be called when a user control has been pressed and an indication of this should be sent to a remote node. If the user control should be repeated, the plugin will automatically send user control repeat messages at fixed intervals according to the plugin configuration. Every call to this function for a repeatable user control should be followed by a call to [emberAfRf4ceMsoUserControlRelease](#).

Parameters

<i>pairingIndex</i>	The pairing index to which to send user control messages.
<i>rcCommand-Code</i>	The RC command code of the user control.
<i>rcCommand-Payload</i>	The optional RC command payload of the user control.
<i>rcCommand-PayloadLength</i>	The length of the optional RC command payload of the user control.
<i>atomic</i>	true if the user control is atomic or false if it should repeat.

This function is only applicable to controller devices.

Returns

An ::EmberStatus value that indicates the success or failure of the command.

6.5.3.7 EmberStatus emberAfRf4ceMsoUserControlRelease (*uint8_t pairingIndex, EmberAfRf4ceMsoKeyCode rcCommandCode*)

Send indications of user control release to a pairing index.

This function can be called when a user control has been released and an indication of this should be sent to a remote node.

Parameters

<i>pairingIndex</i>	The pairing index to which to send user control messages.
<i>rcCommand-Code</i>	The RC command code of the user control.

This function is only applicable to controller devices.

Returns

An ::EmberStatus value that indicates the success or failure of the command.

6.5.3.8 EmberStatus emberAfRf4ceMsoSetAttributeRequest (*uint8_t pairingIndex, EmberAfRf4ceMsoAttributeId attributId, uint8_t index, uint8_t valueLen, const uint8_t * value*)

Set an attribute on a remote node.

This function can be called to set an attribute on a remote node. After sending the request, the plugin will automatically idle for a fixed duration as required by the MSO specification and then wait a configurable duration for the response from the target. The plugin will call ::emberAfPluginRf4ceMsoSetAttribute-ResponseCallback when the response is received or if a timeout occurs.

Parameters

<i>pairingIndex</i>	The pairing index on which to set the attribute.
<i>attributId</i>	The attribute id to set..
<i>index</i>	The index of the element for vector attributes.
<i>valueLen</i>	The length of the value to set.
<i>value</i>	The value to set.

This function is only applicable to controller devices.

Returns

An ::EmberStatus value that indicates the success or failure of the command.

6.5.3.9 EmberStatus emberAfRf4ceMsoGetAttributeRequest (*uint8_t pairingIndex, EmberAfRf4ceMsoAttributeId attributelD, uint8_t index, uint8_t valueLen*)

Get an attribute on a remote node.

This function can be called to get an attribute on a remote node. After sending the request, the plugin will automatically idle for a fixed duration as required by the MSO specification and then wait a configurable duration for the response from the target. The plugin will call ::emberAfPluginRf4ceMsoGetAttributeResponseCallback when the response is received or if a timeout occurs.

Parameters

<i>pairingIndex</i>	The pairing index on which to get the attribute.
<i>attributeId</i>	The attribute id to get.
<i>index</i>	The index of the element for vector attributes.
<i>valueLen</i>	The length of the value to get.

This function is only applicable to controller devices.

Returns

An ::EmberStatus value that indicates the success or failure of the command.

6.5.3.10 bool emberAfRf4ceMsolrRfDatabaseEntryUseDefault (*const EmberAfRf4ceMsoIrRfDatabaseEntry * entry*)

Determine if the IR-RF database entry is the default.

Implemented as a macro for efficiency.

Returns

true if the IR-RF database table entry is the default or false otherwise.

6.5.3.11 bool emberAfRf4ceMsolrRfDatabaseEntryHasRfPressedDescriptor (*const EmberAfRf4ceMsoIrRfDatabaseEntry * entry*)

Determine if the IR-RF database entry has an RF pressed descriptor.

Implemented as a macro for efficiency.

Returns

true if the IR-RF database table entry has an RF pressed descriptor or false otherwise.

6.5.3.12 bool emberAfRf4ceMsolrRfDatabaseEntryHasRfRepeatedDescriptor (const EmberAfRf4ceMsoIrRfDatabaseEntry * entry)

Determine if the IR-RF database entry has an RF repeated descriptor.

Implemented as a macro for efficiency.

Returns

true if the IR-RF database table entry has an RF repeated descriptor or false otherwise.

6.5.3.13 bool emberAfRf4ceMsolrRfDatabaseEntryHasRfReleasedDescriptor (const EmberAfRf4ceMsoIrRfDatabaseEntry * entry)

Determine if the IR-RF database entry has an RF released descriptor.

Implemented as a macro for efficiency.

Returns

true if the IR-RF database table entry has an RF released descriptor or false otherwise.

6.5.3.14 uint8_t emberAfRf4ceMsolrRfDatabaseEntryGetMinimumNumberOfTransmissions (const EmberAfRf4ceMsoIrRfDatabaseRfDescriptor * rfDescriptor)

Get the minimum number of transmissions for a key code from its RF RF descriptor.

Implemented as a macro for efficiency.

Returns

The minimum number of transmissions.

6.5.3.15 bool emberAfRf4ceMsolrRfDatabaseEntryShouldTransmitUntilRelease (const EmberAfRf4ceMsoIrRfDatabaseRfDescriptor * rfDescriptor)

Determine if a key code should continue being transmitted after the minimum number of transmissions have taken place when the key is kept pressed.

Implemented as a macro for efficiency.

Returns

true if the key code should continue being transmitted after the minimum number of transmissions have taken place or false otherwise.

6.5.3.16 bool emberAfRf4ceMsolrRfDatabaseEntryHasIrDescriptor (const EmberAfRf4ceMsoIrRfDatabaseEntry * entry)

Determine if the IR-RF database entry has an IR descriptor.

Implemented as a macro for efficiency.

Returns

true if the IR-RF database table entry has an IR descriptor or false otherwise.

6.6 RF4CE Multiple System Operators IR-RF Database Originator

Functions

- EmberStatus `emberAfRf4ceMsoIrRfDatabaseOriginatorGet` (`EmberAfRf4ceMsoKeyCode` `keyCode`, `EmberAfRf4ceMsoIrRfDatabaseEntry` *`entry`)
- EmberStatus `emberAfRf4ceMsoIrRfDatabaseOriginatorSet` (`EmberAfRf4ceMsoKeyCode` `keyCode`, const `EmberAfRf4ceMsoIrRfDatabaseEntry` *`entry`)
- EmberStatus `emberAfRf4ceMsoIrRfDatabaseOriginatorClear` (`EmberAfRf4ceMsoKeyCode` `keyCode`)
- void `emberAfRf4ceMsoIrRfDatabaseOriginatorClearAll` (void)

6.6.1 Detailed Description

The RF4CE Multiple System Operators (MSO) IR-RF Database Originator plugin implements the optional IR-RF database feature of the MSO profile for controllers. The IR-RF database provides a standard mechanism for remapping keys on a remote to control legacy IR devices or to perform simultaneous IR and RF functions. This plugin manages the storage and retrieval of these mappings for controllers.

When the application queries the target for IR-RF information via the [RF4CE Multiple System Operators Profile](#) plugin, the data from the target are passed to this plugin via [emberAfPluginRf4ceMsoIncomingIrRfDatabaseAttributeCallback](#). This plugin, in turn, will store the data in RAM. The keys that may be remapped are configuration in AppBuilder and the amount of storage space dedicated to storing IR-RF database entries is configurable via the plugin options.

When [emberAfRf4ceMsoUserControlPress](#) and [emberAfRf4ceMsoUserControlRelease](#) are called, the [RF4CE Multiple System Operators Profile](#) plugin will ask this plugin to retrieve RF mappings for the key via [emberAfPluginRf4ceMsoGetIrRfDatabaseEntryCallback](#). If a database entry exists for the key, it will be used in lieu of the original key. The same callback may be used by the application to retrieve IR mappings for keys.

Support for the optional IR-RF database feature for targets is provided by the [RF4CE Multiple System Operators IR-RF Database Recipient](#) plugin.

6.6.2 Function Documentation

6.6.2.1 EmberStatus `emberAfRf4ceMsoIrRfDatabaseOriginatorGet` (`EmberAfRf4ceMsoKeyCode` `keyCode`, `EmberAfRf4ceMsoIrRfDatabaseEntry` * `entry`)

Get the IR-RF entry from the database for a key code.

Parameters

<code>keyCode</code>	The key code.
<code>entry</code>	A pointer to the EmberAfRf4ceMsoIrRfDatabaseEntry to be populated.

Returns

An ::EmberStatus value that indicates the success or failure of the command.

6.6.2.2 EmberStatus emberAfRf4ceMsolrRfDatabaseOriginatorSet (EmberAfRf4ceMsoKeyCode keyCode, const EmberAfRf4ceMsoIrRfDatabaseEntry * entry)

Set the IR-RF entry in the database for a key code.

Parameters

<i>keyCode</i>	The key code.
<i>entry</i>	A pointer to the EmberAfRf4ceMsoIrRfDatabaseEntry .

Returns

An ::EmberStatus value that indicates the success or failure of the command.

6.6.2.3 EmberStatus emberAfRf4ceMsolrRfDatabaseOriginatorClear (EmberAfRf4ceMsoKeyCode keyCode)

Clear the IR-RF entry from the database for a key code.

Parameters

<i>keyCode</i>	The key code.
----------------	---------------

Returns

An ::EmberStatus value that indicates the success or failure of the command.

6.6.2.4 void emberAfRf4ceMsolrRfDatabaseOriginatorClearAll (void)

Clear all of the IR-RF entries from the database.

6.7 RF4CE Multiple System Operators IR-RF Database Recipient

Functions

- EmberStatus `emberAfRf4ceMsoIrRfDatabaseRecipientAdd` (`EmberAfRf4ceMsoKeyCode keyCode, EmberAfRf4ceMsoIrRfDatabaseEntry *entry`)
- EmberStatus `emberAfRf4ceMsoIrRfDatabaseRecipientRemove` (`EmberAfRf4ceMsoKeyCode keyCode`)
- void `emberAfRf4ceMsoIrRfDatabaseRecipientRemoveAll` (`void`)

6.7.1 Detailed Description

The RF4CE Multiple System Operators (MSO) IR-RF Database Recipient plugin implements the optional IR-RF database feature of the MSO profile for targets. The IR-RF database provides a standard mechanism for remapping keys on a remote to control legacy IR devices or to perform simultaneous IR and RF functions. This plugin manages the storage and retrieval of these mappings for targets.

When a controller queries this device for IR-RF information, the [RF4CE Multiple System Operators Profile](#) plugin will pass the request to this plugin via `emberAfPluginRf4ceMsoHaveIrRfDatabaseAttributeCallback` and `emberAfPluginRf4ceMsoGetIrRfDatabaseAttributeCallback`. This plugin, in turn, will provide the IR-RF information so that it may be sent back to the controller.

This plugin only stores IR-RF database entries on behalf of the application. The actual IR and RF descriptors themselves must be provided by the application. Entries can be added to the database by calling `emberAfRf4ceMsoIrRfDatabaseRecipientAdd` with the key code to be remapped. `emberAfRf4ceMsoIrRfDatabaseRecipientRemove` can be used to clear a specific entry or `emberAfRf4ceMsoIrRfDatabaseRecipientRemoveAll` may be used to clear all entries in the database. Note that database entries are stored in RAM. The amount of storage space dedicated to storing IR-RF database entries is configurable via the plugin options.

Support for the optional IR-RF database feature for controllers is provided by the [RF4CE Multiple System Operators IR-RF Database Originator](#) plugin.

6.7.2 Function Documentation

6.7.2.1 EmberStatus `emberAfRf4ceMsoIrRfDatabaseRecipientAdd` (`EmberAfRf4ceMsoKeyCode keyCode, EmberAfRf4ceMsoIrRfDatabaseEntry * entry`)

Add an IR-RF entry to the database for a key code.

Parameters

<code>keyCode</code>	The key code.
<code>entry</code>	A pointer to the <code>EmberAfRf4ceMsoIrRfDatabaseEntry</code> .

Returns

An ::EmberStatus value that indicates the success or failure of the command.

**6.7.2.2 EmberStatus emberAfRf4ceMsolrRfDatabaseRecipientRemove (EmberAfRf4ceMsoKeyCode
keyCode)**

Remove an IR-RF entry from the database.

Parameters

<i>keyCode</i>	The key code which entry is to be removed.
----------------	--

Returns

An ::EmberStatus value that indicates the success or failure of the command.

6.7.2.3 void emberAfRf4ceMsolrRfDatabaseRecipientRemoveAll (void)

Clear all of the IR-RF entries from the database.

6.8 RF4CE Profile Support

Macros

- #define EMBER_AF_RF4CE_MAXIMUM_RF4CE_PAYLOAD_LENGTH
- #define EMBER_AF_RF4CE_MESSAGE_TAG_MASK

Functions

- bool `emberAfRf4ceIsCurrentNetwork` (void)
- EmberStatus `emberAfRf4cePushNetworkIndex` (void)
- EmberStatus `emberAfRf4ceStart` (void)
- EmberStatus `emberAfRf4ceSetPowerSavingParameters` (uint32_t dutyCycleMs, uint32_t activePeriodMs)
- EmberStatus `emberAfRf4ceRxEnable` (EmberAfRf4ceProfileId profileId, bool enable)
- EmberStatus `emberAfRf4ceSetFrequencyAgilityParameters` (uint8_t rssWindowSz, uint8_t channelChangeReads, int8_t rssThreshold, uint16_t readIntervalS, uint8_t readDuration)
- EmberStatus `emberAfRf4ceSetDiscoveryLqiThreshold` (uint8_t threshold)
- uint8_t `emberAfRf4ceGetBaseChannel` (void)
- EmberStatus `emberAfRf4ceDiscovery` (EmberPanId panId, EmberNodeId nodeId, uint8_t searchDevType, uint16_t discDurationMs, uint8_t maxDiscRepetitions, uint8_t discProfileIdListLength, uint8_t *discProfileIdList)
- EmberStatus `emberAfRf4ceEnableAutoDiscoveryResponse` (uint16_t durationMs, uint8_t discProfileIdListLength, uint8_t *discProfileIdList)
- EmberStatus `emberAfRf4cePair` (uint8_t channel, EmberPanId panId, EmberEUI64 ieeeAddr, uint8_t keyExchangeTransferCount, `EmberAfRf4cePairCompleteCallback` pairCompleteCallback)
- uint8_t `emberAfRf4ceGetPairingIndex` (void)
- EmberStatus `emberAfRf4ceSetPairingTableEntry` (uint8_t pairingIndex, EmberRf4cePairingTableEntry *entry)
- EmberStatus `emberAfRf4ceGetPairingTableEntry` (uint8_t pairingIndex, EmberRf4cePairingTableEntry *entry)
- EmberStatus `emberAfRf4ceSetApplicationInfo` (EmberRf4ceApplicationInfo *appInfo)
- EmberStatus `emberAfRf4ceGetApplicationInfo` (EmberRf4ceApplicationInfo *appInfo)
- EmberStatus `emberAfRf4ceKeyUpdate` (uint8_t pairingIndex, EmberKeyData *key)
- EmberStatus `emberAfRf4ceSend` (uint8_t pairingIndex, uint8_t profileId, uint8_t *message, uint8_t messageLength, uint8_t *messageTag)
- EmberStatus `emberAfRf4ceSendVendorSpecific` (uint8_t pairingIndex, uint8_t profileId, uint16_t vendorId, uint8_t *message, uint8_t messageLength, uint8_t *messageTag)
- EmberStatus `emberAfRf4ceSendExtended` (uint8_t pairingIndex, uint8_t profileId, uint16_t vendorId, EmberRf4ceTxOption txOptions, uint8_t *message, uint8_t messageLength, uint8_t *messageTag)
- EmberStatus `emberAfRf4ceGetDefaultTxOptions` (uint8_t pairingIndex, EmberRf4ceTxOption *txOptions)
- EmberStatus `emberAfRf4ceUnpair` (uint8_t pairingIndex)
- EmberStatus `emberAfRf4ceStop` (void)
- uint8_t `emberAfRf4ceGetMaxPayload` (uint8_t pairingIndex, EmberRf4ceTxOption txOptions)
- uint8_t `emberAfRf4ceDeviceTypeListLength` (EmberRf4ceApplicationCapabilities capabilities)
- uint8_t `emberAfRf4ceProfileIdListLength` (EmberRf4ceApplicationCapabilities capabilities)
- uint16_t `emberAfRf4ceVendorId` (void)
- bool `emberAfRf4ceIsDeviceTypeSupported` (const EmberRf4ceApplicationInfo *appInfo, `EmberAfRf4ceDeviceType` deviceType)

- bool `emberAfRf4ceIsDeviceTypeSupportedLocally` (`EmberAfRf4ceDeviceType` deviceType)
- bool `emberAfRf4ceIsProfileSupported` (const `EmberRf4ceApplicationInfo` *appInfo, `EmberAfRf4ceProfileId` profileId)
- bool `emberAfRf4ceIsProfileSupportedLocally` (`EmberAfRf4ceProfileId` profileId)
- bool `emberAfRf4cePairingTableEntryIsUnused` (const `EmberRf4cePairingTableEntry` *pairingTableEntry)
- bool `emberAfRf4cePairingTableEntryIsProvisional` (const `EmberRf4cePairingTableEntry` *pairingTableEntry)
- bool `emberAfRf4cePairingTableEntryIsActive` (const `EmberRf4cePairingTableEntry` *pairingTableEntry)
- bool `emberAfRf4cePairingTableEntryHasLinkKey` (const `EmberRf4cePairingTableEntry` *pairingTableEntry)
- bool `emberAfRf4cePairingTableEntryIsPairingInitiator` (const `EmberRf4cePairingTableEntry` *pairingTableEntry)
- bool `emberAfRf4cePairingTableEntryHasSecurity` (const `EmberRf4cePairingTableEntry` *pairingTableEntry)
- bool `emberAfRf4cePairingTableEntryHasChannelNormalization` (const `EmberRf4cePairingTableEntry` *pairingTableEntry)

6.8.1 Detailed Description

The RF4CE Profile Support plugin provides the necessary foundation of APIs to interface with an RF4CE-capable device.

The functionality contained in this plugin provides basic RF4CE networking features like discovery, pairing, security, profile and device support, and transmission. In order to commence operations in an RF4CE network, one must call `emberAfRf4ceStart`. This is the starting point for beginning any RF4CE activity, over any profile. In the same sense, RF4CE network operations can be stopped with a call to `emberAfRf4ceStop`.

After network operations have been brought up as discussed above, one can easily configure their device using APIs in this plugin, regardless of what profile they are operating on. See `emberAfRf4ceSetPowerSavingParameters`, `emberAfRf4ceRxEnable`, `emberAfRf4ceSetDiscoveryLqiThreshold`, and `emberAfRf4ceSetApplicationInfo` for examples of ways to do this.

Once an RF4CE network has been started as described above, one can use the `::emberAfDiscovery` and `emberAfRf4cePair` APIs to initiate discovery and pairing processes, respectively. This functionality is complemented by various other discovery and pairing helper functions that aid in configuring the mechanisms. For example, a device can call `emberAfRf4ceEnableAutoDiscoveryResponse` to have the plugin handle discovery request messages. There are also convenience macros to read different information regarding the pairing table entries.

Note that in this plugin, these functions are purposely very generic. Please see [Generic Device Profile](#), [ZigBee Remote Control 1.1 Profile](#), [ZigBee Remote Control 2.0 Profile](#), and [RF4CE Multiple System Operators Profile](#) for specific implementations of RF4CE profiles.

6.8.2 Macro Definition Documentation

6.8.2.1 #define EMBER_AF_RF4CE_MAXIMUM_RF4CE_PAYLOAD_LENGTH

Definition at line 720 of file `rf4ce-profile.h`.

6.8.2.2 #define EMBER_AF_RF4CE_MESSAGE_TAG_MASK

The mask applied by `emberAfRf4ceSend`, `emberAfRf4ceSendVendorSpecific`, and `emberAfRf4ceSendExtended` when allocating a new message tag to an outgoing message. Customers who call `::emberRf4ceSend` or `::ezspRf4ceSend` directly must use message tags outside this mask.

Definition at line 728 of file `rf4ce-profile.h`.

6.8.3 Function Documentation

6.8.3.1 bool emberAfRf4celsCurrentNetwork (void)

Determine if the current network is the RF4CE network.

Returns

true if the current network is the RF4CE network or false otherwise.

6.8.3.2 EmberStatus emberAfRf4cePushNetworkIndex (void)

Set the current network to the RF4CE network.

This function is a convenience wrapper for `::emberAfPushNetworkIndex`. Like the other push APIs, every call to this API must be paired with a subsequent call to `::emberAfPopNetworkIndex`. Note that it is not necessary to call function before any of the `emberAfRf4ce` functions or in any `emberAfPluginRf4ce` callbacks. This function is intended primarily for internal use, but is made available to the application in case it is useful for application-specific behavior not generally supported by the plugin.

Returns

An `:EmberStatus` value that indicates the success or failure of the command.

6.8.3.3 EmberStatus emberAfRf4ceStart (void)

Start the RF4CE network operations.

The function is a convenience wrapper for `::emberRf4ceStart` and `::ezspRf4ceStart`. It will start the network operations according to the plugin configuration.

Note that this function automatically switches to the RF4CE network index before calling any stack APIs and switches back to the previous network index when it is finished. Therefore, it is not necessary to push and pop the RF4CE network index before and after calling this function.

Returns

An `:EmberStatus` value that indicates the success or failure of the command.

6.8.3.4 EmberStatus emberAfRf4ceSetPowerSavingParameters (uint32_t *dutyCycleMs*, uint32_t *activePeriodMs*)

Set the power-saving parameters.

Setting the duty cycle to zero disables power saving and will force the receiver to be kept on. It is equivalent to using `emberAfRf4ceRxEnable` to set the wildcard profile to enabled. Setting the duty cycle to non-zero and the active period to zero will allow the receiver to be disabled when all of the profiles are inactive. It is equivalent to setting the wildcard profile to disabled. Otherwise, when all profiles are inactive, the receiver will duty cycle, will the receiver turned on for `activePeriodMs` within each `dutyCycleMs` period.

Note that this function automatically switches to the RF4CE network index before calling any stack APIs and switches back to the previous network index when it is finished. Therefore, it is not necessary to push and pop the RF4CE network index before and after calling this function.

Returns

An `:EmberStatus` value that indicates the success or failure of the command.

6.8.3.5 EmberStatus emberAfRf4ceRxEnable (`EmberAfRf4ceProfileId profileId, bool enable`)

Enable or disable the receiver for the given profile.

Each profile can individually indicate whether it needs the receiver on or if the receiver can be disabled to conserve power. If at least one profile requires the receiver to be on (e.g., when it expects to receive a message), the plugin will enable the receiver. Otherwise, the device will revert back to the previously specified power-saving parameters.

The profile `EMBER_AF_RF4CE_PROFILE_WILDCARD` may be used to set the default state for the application. Enabling the receiver for the wildcard profile will force the receiver to be kept on. Disabling the receiver for the wildcard profile will permit the receiver to be turned off when all of the other profiles are inactive.

Note that this function automatically switches to the RF4CE network index before calling any stack APIs and switches back to the previous network index when it is finished. Therefore, it is not necessary to push and pop the RF4CE network index before and after calling this function.

Returns

An `:EmberStatus` value that indicates the success or failure of the command.

6.8.3.6 EmberStatus emberAfRf4ceSetFrequencyAgilityParameters (`uint8_t rssiWindowSize, uint8_t channelChangeReads, int8_t rssiThreshold, uint16_t readIntervals, uint8_t readDuration`)

Set the frequency agility parameters.

The function is a convenience wrapper for `:emberRf4ceSetFrequencyAgilityParameters` and `:ezspRf4ceSetFrequencyAgilityParameters`.

Note that this function automatically switches to the RF4CE network index before calling any stack APIs and switches back to the previous network index when it is finished. Therefore, it is not necessary to push and pop the RF4CE network index before and after calling this function.

Returns

An `:EmberStatus` value that indicates the success or failure of the command.

6.8.3.7 EmberStatus emberAfRf4ceSetDiscoveryLqiThreshold (`uint8_t threshold`)

Set the discovery LQI threshold.

The function is a convenience wrapper for ::emberRf4ceSetDiscoveryLqiThreshold and ::ezspSetValue(EZSP_VALUE_RF4CE_DISCOVERY_LQI_THRESHOLD, ...).

Note that this function automatically switches to the RF4CE network index before calling any stack APIs and switches back to the previous network index when it is finished. Therefore, it is not necessary to push and pop the RF4CE network index before and after calling this function.

Returns

An ::EmberStatus value that indicates the success or failure of the command.

6.8.3.8 uint8_t emberAfRf4ceGetBaseChannel (`void`)

Get the device RF4CE base channel.

The function is a convenience wrapper for ::emberRf4ceGetBaseChannel and ::ezspGetValue(EZSP_VALUE_RF4CE_BASE_CHANNEL, ...).

Note that this function automatically switches to the RF4CE network index before calling any stack APIs and switches back to the previous network index when it is finished. Therefore, it is not necessary to push and pop the RF4CE network index before and after calling this function.

Returns

The device RF4CE base channel.

6.8.3.9 EmberStatus emberAfRf4ceDiscovery (`EmberPanId panId, EmberNodeID nodeId,` `uint8_t searchDevType, uint16_t discDurationMs, uint8_t maxDiscRepetitions, uint8_t` `discProfileIDListLength, uint8_t * discProfileIDList`)

Start the discovery procedure.

The function is a convenience wrapper for ::emberRf4ceDiscovery and ::ezspRf4ceDiscovery. The plugin will call the DiscoveryResponse callback for any profile id that was requested if a response is received from a node supporting that profile. For any given response, discovery will continue if at least one matching profile requests it. Otherwise, the plugin instructs the stack to end discovery at the conclusion of the current triad. When discovery completes, the plugin will call the DiscoveryComplete callback for each profile id that was requested.

Note that this function automatically switches to the RF4CE network index before calling any stack APIs and switches back to the previous network index when it is finished. Therefore, it is not necessary to push and pop the RF4CE network index before and after calling this function.

Returns

An ::EmberStatus value that indicates the success or failure of the command.

6.8.3.10 EmberStatus emberAfRf4ceEnableAutoDiscoveryResponse (`uint16_t durationMs, uint8_t` `discProfileIDListLength, uint8_t * discProfileIDList`)

Enable auto discovery mode.

The function is a convenience wrapper for ::emberRf4ceEnableAutoDiscoveryResponse and ::ezspRf4ceEnableAutoDiscoveryResponse. When auto discovery completes, the plugin will call the AutoDiscoveryResponseComplete callback for each profile id that was requested.

Note that this function automatically switches to the RF4CE network index before calling any stack APIs and switches back to the previous network index when it is finished. Therefore, it is not necessary to push and pop the RF4CE network index before and after calling this function.

Returns

An ::EmberStatus value that indicates the success or failure of the command.

6.8.3.11 EmberStatus emberAfRf4cePair (uint8_t channel, EmberPanId panId, EmberEUI64 ieeeAddr, uint8_t keyExchangeTransferCount, EmberAfRf4cePairCompleteCallback pairCompleteCallback)

Start the pairing procedure.

The function is a convenience wrapper for ::emberRf4cePair and ::ezspRf4cePair. When pairing completes, if a callback was specified in the API call (that is, callback was a non-NULL function pointer), the passed callback shall be called, otherwise the plugin will call the PairComplete callback for each profile id.

Note that this function automatically switches to the RF4CE network index before calling any stack APIs and switches back to the previous network index when it is finished. Therefore, it is not necessary to push and pop the RF4CE network index before and after calling this function.

Returns

An ::EmberStatus value that indicates the success or failure of the command.

6.8.3.12 uint8_t emberAfRf4ceGetPairingIndex (void)

Get the pairing index of the incoming or sent message.

The function returns the pairing index of the sender of the current incoming message or the destination of the current outgoing message. This function can only be called in the context of [emberRf4ceIncomingMessageHandler](#), [ezspRf4ceIncomingMessageHandler](#), [emberRf4ceMessageSentHandler](#), or [ezspRf4ceMessageSentHandler](#).

Note that this function automatically switches to the RF4CE network index before calling any stack APIs and switches back to the previous network index when it is finished. Therefore, it is not necessary to push and pop the RF4CE network index before and after calling this function.

Returns

The pairing index of the incoming or outgoing message or 0xFF if called from outside the context of [emberRf4ceIncomingMessageHandler](#), [ezspRf4ceIncomingMessageHandler](#), [emberRf4ceMessageSentHandler](#), or [ezspRf4ceMessageSentHandler](#).

6.8.3.13 EmberStatus emberAfRf4ceSetPairingTableEntry (uint8_t pairingIndex, EmberRf4cePairingTableEntry * entry)

Set the pairing table entry at a particular index.

The function is a convenience wrapper for ::emberRf4ceSetPairingTableEntry and ::ezspRf4ceSetPairingTableEntry.

Note that this function automatically switches to the RF4CE network index before calling any stack APIs and switches back to the previous network index when it is finished. Therefore, it is not necessary to push and pop the RF4CE network index before and after calling this function.

Returns

An ::EmberStatus value that indicates the success or failure of the command.

6.8.3.14 EmberStatus emberAfRf4ceGetPairingTableEntry (uint8_t *pairingIndex*, EmberRf4cePairingTableEntry * *entry*)

Get the pairing table entry at a particular index.

The function is a convenience wrapper for ::emberRf4ceGetPairingTableEntry and ::ezspRf4ceGetPairingTableEntry.

Note that this function automatically switches to the RF4CE network index before calling any stack APIs and switches back to the previous network index when it is finished. Therefore, it is not necessary to push and pop the RF4CE network index before and after calling this function.

Returns

An ::EmberStatus value that indicates the success or failure of the command.

6.8.3.15 EmberStatus emberAfRf4ceSetApplicationInfo (EmberRf4ceApplicationInfo * *appInfo*)

Set the node application information.

The function is a convenience wrapper for ::emberRf4ceSetApplicationInfo and ::ezspRf4ceSetApplicationInfo.

Note that this function automatically switches to the RF4CE network index before calling any stack APIs and switches back to the previous network index when it is finished. Therefore, it is not necessary to push and pop the RF4CE network index before and after calling this function.

Returns

An ::EmberStatus value that indicates the success or failure of the command.

6.8.3.16 EmberStatus emberAfRf4ceGetApplicationInfo (EmberRf4ceApplicationInfo * *appInfo*)

Get the node application information.

The function is a convenience wrapper for ::emberRf4ceGetApplicationInfo and ::ezspRf4ceGetApplicationInfo.

Note that this function automatically switches to the RF4CE network index before calling any stack APIs and switches back to the previous network index when it is finished. Therefore, it is not necessary to push and pop the RF4CE network index before and after calling this function.

Returns

An ::EmberStatus value that indicates the success or failure of the command.

6.8.3.17 EmberStatus emberAfRf4ceKeyUpdate (*uint8_t pairingIndex, EmberKeyData * key*)

Update the link key of a pairing table entry.

The function is a convenience wrapper for ::emberRf4ceKeyUpdate and ::ezspRf4ceKeyUpdate.

Note that this function automatically switches to the RF4CE network index before calling any stack APIs and switches back to the previous network index when it is finished. Therefore, it is not necessary to push and pop the RF4CE network index before and after calling this function.

Returns

An ::EmberStatus value that indicates the success or failure of the command.

6.8.3.18 EmberStatus emberAfRf4ceSend (*uint8_t pairingIndex, uint8_t profileId, uint8_t * message, uint8_t messageLength, uint8_t * messageTag*)

Send a message to a pairing index.

The function is a convenience wrapper for ::emberRf4ceSend and ::ezspRf4ceSend. If the pairing index is ::EMBER_RF4CE_PAIRING_TABLE_BROADCAST_INDEX, the plugin will broadcast the message. Otherwise, the plugin will send a unicast. For unicasts, the plugin automatically enables security if the remote node is security capable and if a link key exists to the node. If the local node is a target and the remote node supports channel normalization, the plugin will automatically set the channel designator option in the outgoing message. The plugin always requests acknowledgements for unicast messages.

The plugin will allocate a new message tag for the message and, if messageTag is not NULL, store it in the address pointed to by messageTag. See [EMBER_AF_RF4CE_MESSAGE_TAG_MASK](#) for more information about the message tags allocated by the plugin.

Note that this function automatically switches to the RF4CE network index before calling any stack APIs and switches back to the previous network index when it is finished. Therefore, it is not necessary to push and pop the RF4CE network index before and after calling this function.

Returns

An ::EmberStatus value that indicates the success or failure of the command.

6.8.3.19 EmberStatus emberAfRf4ceSendVendorSpecific (*uint8_t pairingIndex, uint8_t profileId, uint16_t vendorId, uint8_t * message, uint8_t messageLength, uint8_t * messageTag*)

Send a vendor-specific message to a pairing index.

The function is a convenience wrapper for ::emberRf4ceSend and ::ezspRf4ceSend. If the pairing index is ::EMBER_RF4CE_PAIRING_TABLE_BROADCAST_INDEX, the plugin will broadcast the message. Otherwise, the plugin will send a unicast. For unicasts, the plugin automatically enables security if the remote node is security capable and if a link key exists to the node. If the local node is a target and the remote node supports channel normalization, the plugin will automatically set the channel designator option in the outgoing message. The plugin always requests acknowledgements for unicast messages.

The plugin will allocate a new message tag for the message and, if messageTag is not NULL, store it in the address pointed to by messageTag. See [EMBER_AF_RF4CE_MESSAGE_TAG_MASK](#) for more information about the message tags allocated by the plugin.

Note that this function automatically switches to the RF4CE network index before calling any stack APIs and switches back to the previous network index when it is finished. Therefore, it is not necessary to push and pop the RF4CE network index before and after calling this function.

Returns

An ::EmberStatus value that indicates the success or failure of the command.

6.8.3.20 EmberStatus emberAfRf4ceSendExtended (*uint8_t pairingIndex, uint8_t profileId, uint16_t vendorId, EmberRf4ceTxOption txOptions, uint8_t * message, uint8_t messageLength, uint8_t * messageTag*)

Send a message to a pairing index specifying the transmission options as described in the RF4CE specifications.

The function is a convenience wrapper for ::emberRf4ceSend and ::ezspRf4ceSend. If the pairing index is ::EMBER_RF4CE_PAIRING_TABLE_BROADCAST_INDEX, the plugin will broadcast the message. Otherwise, the plugin will send a unicast.

The plugin will allocate a new message tag for the message and, if messageTag is not NULL, store it in the address pointed to by messageTag. See [EMBER_AF_RF4CE_MESSAGE_TAG_MASK](#) for more information about the message tags allocated by the plugin.

Note that this function automatically switches to the RF4CE network index before calling any stack APIs and switches back to the previous network index when it is finished. Therefore, it is not necessary to push and pop the RF4CE network index before and after calling this function.

Returns

An ::EmberStatus value that indicates the success or failure of the command.

6.8.3.21 EmberStatus emberAfRf4ceGetDefaultTxOptions (*uint8_t pairingIndex, EmberRf4ceTxOption * txOptions*)

Get the default TX options for the pairing index.

Note that this function automatically switches to the RF4CE network index before calling any stack APIs and switches back to the previous network index when it is finished. Therefore, it is not necessary to push and pop the RF4CE network index before and after calling this function.

Parameters

<i>pairingIndex</i>	The pairing index to check.
<i>txOptions</i>	A pointer to the ::EmberRf4ceTxOption to be populated.

Returns

An ::EmberStatus value that indicates the success or failure of the command.

6.8.3.22 EmberStatus emberAfRf4ceUnpair (*uint8_t pairingIndex*)

Remove a pairing.

The function is a convenience wrapper for ::emberRf4ceUnpair and ::ezspRf4ceUnpair. When unpairing completes, the plugin will call UnpairComplete callback for each profile id.

Note that this function automatically switches to the RF4CE network index before calling any stack APIs and switches back to the previous network index when it is finished. Therefore, it is not necessary to push and pop the RF4CE network index before and after calling this function.

Returns

An `:EmberStatus` value that indicates the success or failure of the command.

6.8.3.23 `EmberStatus emberAfRf4ceStop (void)`

Stop the RF4CE network operations.

The function is a convenience wrapper for `::emberRf4ceStop` and `::ezspRf4ceStop`.

Note that this function automatically switches to the RF4CE network index before calling any stack APIs and switches back to the previous network index when it is finished. Therefore, it is not necessary to push and pop the RF4CE network index before and after calling this function.

Returns

An `:EmberStatus` value that indicates the success or failure of the command.

6.8.3.24 `uint8_t emberAfRf4ceGetMaxPayload (uint8_t pairingIndex, EmberRf4ceTxOption txOptions)`

Returns the maximum RF4CE network layer payload.

The function is a convenience wrapper for `::emberRf4ceGetMaxPayload` and `::ezspRf4ceGetMaxPayload`.

Note that this function automatically switches to the RF4CE network index before calling any stack APIs and switches back to the previous network index when it is finished. Therefore, it is not necessary to push and pop the RF4CE network index before and after calling this function.

Returns

The maximum allowed length in bytes of the RF4CE network layer payload according to the passed pairing index and TX options.

6.8.3.25 `uint8_t emberAfRf4ceDeviceTypeListLength (EmberRf4ceApplicationCapabilities capabilities)`

Get the length of the device type list from the application capabilities bitmask.

Implemented as a macro for efficiency.

Returns

The length of the device type list.

6.8.3.26 `uint8_t emberAfRf4ceProfileIdListLength (EmberRf4ceApplicationCapabilities capabilities)`

Get the length of the profile id list from the application capabilities bitmask.

Implemented as a macro for efficiency.

Returns

The length of the profile id list.

6.8.3.27 uint16_t emberAfRf4ceVendorId (void)

Get the vendor id of the local application.

Implemented as a macro for efficiency.

Returns

The vendor id.

6.8.3.28 bool emberAfRf4ceIsDeviceTypeSupported (const EmberRf4ceApplicationInfo * appInfo, EmberAfRf4ceDeviceType deviceType)

Determine if the application implements a particular device type.

If the device type is [EMBER_AF_RF4CE_DEVICE_TYPE_WILDCARD](#), this function always returns true.

Returns

true if the application implements the device type or false otherwise.

6.8.3.29 bool emberAfRf4ceIsDeviceTypeSupportedLocally (EmberAfRf4ceDeviceType deviceType)

Determine if the local application implements a particular device type.

If the device type is [EMBER_AF_RF4CE_DEVICE_TYPE_WILDCARD](#), this function always returns true.

Implemented as a macro for efficiency.

Returns

true if the application implements the device type or false otherwise.

6.8.3.30 bool emberAfRf4ceIsProfileSupported (const EmberRf4ceApplicationInfo * appInfo, EmberAfRf4ceProfileId profileId)

Determine if the application implements a particular profile.

Returns

true if the application implements the profile or false otherwise.

6.8.3.31 bool emberAfRf4ceIsProfileSupportedLocally (EmberAfRf4ceProfileId *profileId*)

Determine if the local application implements a particular profile.

Implemented as a macro for efficiency.

Returns

true if the application implements the profile or false otherwise.

6.8.3.32 bool emberAfRf4ceIsPairingTableEntryUnused (const EmberRf4cePairingTableEntry * *pairingTableEntry*)

Determine if the pairing table entry is unused.

Implemented as a macro for efficiency.

Returns

true if the pairing table entry is unused or false otherwise.

6.8.3.33 bool emberAfRf4ceIsPairingTableEntryProvisional (const EmberRf4cePairingTableEntry * *pairingTableEntry*)

Determine if the pairing table entry is provisional.

Implemented as a macro for efficiency.

Returns

true if the pairing table entry is provisional or false otherwise.

6.8.3.34 bool emberAfRf4ceIsPairingTableEntryActive (const EmberRf4cePairingTableEntry * *pairingTableEntry*)

Determine if the pairing table entry is active.

Implemented as a macro for efficiency.

Returns

true if the pairing table entry is active or false otherwise.

6.8.3.35 bool emberAfRf4ceIsPairingTableEntryHasLinkKey (const EmberRf4cePairingTableEntry * *pairingTableEntry*)

Determine if the pairing table entry has a link key.

Implemented as a macro for efficiency.

Returns

true if the pairing table entry has a link key or false otherwise.

6.8.3.36 bool emberAfRf4cePairingTableEntryIsPairingInitiator (const EmberRf4cePairingTableEntry * *pairingTableEntry*)

Determine if the local node was the pairing initiator for the pairing table entry.

Implemented as a macro for efficiency.

Returns

true if the local node was the pairing initiator for the pairing table entry or false otherwise.

6.8.3.37 bool emberAfRf4cePairingTableEntryHasSecurity (const EmberRf4cePairingTableEntry * *pairingTableEntry*)

Determine if the pairing table entry is security capable.

Implemented as a macro for efficiency.

Returns

true if the pairing table entry is security capable or false otherwise.

6.8.3.38 bool emberAfRf4cePairingTableEntryHasChannelNormalization (const EmberRf4cePairingTableEntry * *pairingTableEntry*)

Determine if the pairing table entry supports channel normalization.

Implemented as a macro for efficiency.

Returns

true if the pairing table entry supports channel normalization or false otherwise.

6.9 ZigBee Remote Control 1.1 Profile

Functions

- EmberStatus `emberAfRf4ceZrc11Discovery` (EmberPanId panId, EmberNodeId nodeId, `EmberAfRf4ceDeviceType` searchDevType)
- EmberStatus `emberAfRf4ceZrc11EnableAutoDiscoveryResponse` (void)
- EmberStatus `emberAfRf4ceZrc11UserControlPress` (uint8_t pairingIndex, `EmberAfRf4ceUserControlCode` rcCommandCode, const uint8_t *rcCommandPayload, uint8_t rcCommandPayloadLength, bool atomic)
- EmberStatus `emberAfRf4ceZrc11UserControlRelease` (uint8_t pairingIndex, `EmberAfRf4ceUserControlCode` rcCommandCode)
- EmberStatus `emberAfRf4ceZrc11CommandDiscoveryRequest` (uint8_t pairingIndex)
- uint8_t * `emberAfRf4ceZrcCommandsSupportedContents` (`EmberAfRf4ceZrcCommandsSupported` *commandsSupported)

6.9.1 Detailed Description

The ZigBee Remote Control 1.1 (ZRC1.1) plugin implements the ZRC1.1 profile. The profile describes a simple discovery and pairing procedure for joining devices and a simple user control procedure for controlling devices via HDMI-CEC UI commands. The plugin manages these procedures for both originators and recipients. Note that this plugin implements version 1.1 of the ZRC profile. Version 1.1 devices are fully interoperable with version 1.0 devices. (ZRC1.0 was also known as the Consumer Electronics Remote Control (CERC) profile.) As required by the specification, this plugin sends ZRC1.1-formatted messages and receives both ZRC1.0- and ZRC1.1- formatted messages.

This plugin supports originators and recipients. Originators send user control messages to recipients after pairing with them. Originators are most typically controllers but may also be targets. Recipients must be targets. Both originators and recipients must start general network operations before beginning ZRC-specific operations. Network operations should be started by calling `emberAfRf4ceStart` in the [RF4CE Profile Support](#) plugin.

Once network operations have started, ZRC1.1 originators can initiate the discovery and pairing procedure by calling `emberAfRf4ceZrc11Discovery` with the appropriate discovery parameters. The plugin will first perform discovery for matching targets in range. If exactly one target is identified, the plugin will then continue to the pairing procedure. At the conclusion of the discovery and pairing procedure, the plugin will call `emberAfPluginRf4ceZrc11PairingCompleteCallback` to indicate whether pairing completed successfully with a target and, if so, which pairing index has been assigned to that target.

Recipients can initiate the discovery and pairing procedure by calling `emberAfRf4ceZrc11EnableAutoDiscoveryResponse`. This will put the node into auto discovery mode for a fixed duration of time. During this time, the stack will automatically respond to discovery requests from originators that match the configuration of this device. If an originator discovers this device, the plugin will then wait for a pair request from the same originator. When the process completes, the plugin will call `emberAfPluginRf4ceZrc11PairingCompleteCallback` to indicate whether pairing completed successfully with an originator and, if so, which pairing index has been assigned to that originator.

Following a successful discovery and pairing, originators may send user control messages to targets by calling `emberAfRf4ceZrc11UserControlPress`. In response, the plugin will transmit a user control press message to the indicated target with the HDMI-CEC command code and payload. If the press is atomic, the plugin performs no additional processing. Otherwise, the plugin will repeatedly transmit user control repeat messages at fixed intervals until `emberAfRf4ceZrc11UserControlRelease` is called. The interval at which user control repeat messages are transmitted is configurable in the plugin options.

For targets, the plugin will keep track of incoming user control messages. Each time a user control press command is received, the plugin will call `emberAfPluginRf4ceZrc11UserControlCallback` with its HDMI-I-CEC command code and payload. The plugin will then wait for a fixed duration for a corresponding user control repeat or release messages. If a user control repeat message is received within the timeout, the plugin will reset its timer and wait for the next message. If a user control repeat message is not received within the timeout or if a user control release is received, the plugin will call `emberAfPluginRf4ceZrc11UserControlCallback` with an indication that the user control has stopped. The timeout for receiving repeat messages is configurable in the plugin options. Note that the plugin will not call `emberAfPluginRf4ceZrc11UserControlCallback` for repeat messages that follow a press message.

The plugin is capable of keeping track of a fixed number of simultaneous incoming and outgoing user control messages. The limits are configurable in the plugin options.

Originators and recipients may each perform command discovery to determine which user control codes are supported by the paired node. This is done by calling `emberAfRf4ceZrc11CommandDiscoveryRequest`. Once command discovery completes, the plugin calls `emberAfPluginRf4ceZrc11CommandDiscoveryResponseCallback` with the results. If this node receives a command discovery request, it will automatically respond with the list of HDMI-CEC command codes that the device supports. The supported command codes are configurable in AppBuilder.

This plugin manages the state of the receiver by calling `emberAfRf4ceRxEnable` using `EMBER_AF_RF4CE_PROFILE_REMOTE_CONTROL_1_1` as the profile id. If the application also wishes to manage the receiver, it should do so using `EMBER_AF_RF4CE_PROFILE_WILDCARD` as the profile id or by calling `emberAfRf4ceSetPowerSavingParameters`.

This plugin utilizes the discovery, pairing, sending and receiving, and power-saving functionality provided by the [RF4CE Profile Support](#) plugin.

6.9.2 Function Documentation

6.9.2.1 EmberStatus emberAfRf4ceZrc11Discovery (EmberPanId *panId*, EmberNodeId *nodeId*, EmberAfRf4ceDeviceType *searchDevType*)

Initiate the pairing procedure for an originator.

The plugin begins the pairing procedure by searching for a potential target with which to pair. If exactly one target is found, the plugin will attempt to pair with it. If no targets or more than one target is found, the pairing will fail, as required by the ZRC1.x specification. The plugin will call `emberAfPluginRf4ceZrc11PairingCompleteCallback` with the results of the pairing procedure.

Parameters

<i>panId</i>	The pan id to search or ::EMBER_RF4CE_BROADCAST_PAN_ID.
<i>nodeId</i>	The node id for which to search or ::EMBER_RF4CE_BROADCAST_ADDRESS.
<i>searchDevType</i>	The device type for which to search or <code>EMBER_AF_RF4CE_DEVICE_TYPE_WILDCARD</code> .

Returns

An ::EmberStatus value that indicates the success or failure of the command.

6.9.2.2 EmberStatus emberAfRf4ceZrc11EnableAutoDiscoveryResponse (void)

Initiate the pairing procedure for a recipient.

The plugin performs the binding procedure by enabling auto discovery mode. The plugin will call [emberAfPluginRf4ceZrc11PairingCompleteCallback](#) with the results of the pairing. This function may only be called on a target device.

Returns

An ::EmberStatus value that indicates the success or failure of the command.

6.9.2.3 EmberStatus emberAfRf4ceZrc11UserControlPress (*uint8_t pairingIndex, EmberAfRf4ceUserControlCode rcCommandCode, const uint8_t * rcCommandPayload, uint8_t rcCommandPayloadLength, bool atomic*)

Send indications of user control presses and repeats to a pairing index.

This function can be called when a user control has been pressed and an indication of this should be sent to a target. If the user control should be repeated, the plugin will automatically send user control repeat messages at fixed intervals according to the plugin configuration. Every call to this function for a repeatable user control should be followed by a call to [emberAfRf4ceZrc11UserControlRelease](#).

Parameters

<i>pairingIndex</i>	The pairing index to which to send user control messages.
<i>rcCommand-Code</i>	The RC command code of the user control.
<i>rcCommand-Payload</i>	The optional RC command payload of the user control.
<i>rcCommand-PayloadLength</i>	The length of the optional RC command payload of the user control.
<i>atomic</i>	true if the user control is atomic or false if it should repeat.

Returns

An ::EmberStatus value that indicates the success or failure of the command.

6.9.2.4 EmberStatus emberAfRf4ceZrc11UserControlRelease (*uint8_t pairingIndex, EmberAfRf4ceUserControlCode rcCommandCode*)

Send indications of user control release to a pairing index.

This function can be called when a user control has been released and an indication of this should be sent to a remote node.

Parameters

<i>pairingIndex</i>	The pairing index to which to send user control messages.
<i>rcCommand-Code</i>	The RC command code of the user control.

Returns

An ::EmberStatus value that indicates the success or failure of the command.

6.9.2.5 EmberStatus emberAfRf4ceZrc11CommandDiscoveryRequest (`uint8_t pairingIndex`)

Send a command discovery request to a pairing index.

This function can be called to discover which commands are supported by a remote node. The plugin will call [emberAfPluginRf4ceZrc11CommandDiscoveryResponseCallback](#) when the response is received.

Parameters

<code>pairingIndex</code>	The pairing index to which to the request.
---------------------------	--

Returns

An `:EmberStatus` value that indicates the success or failure of the command.

6.9.2.6 `uint8_t* emberAfRf4ceZrcCommandsSupportedContents (EmberAfRf4ceZrcCommands-Supported * commandsSupported)`

Accesses the actual command discovery data of the [EmberAfRf4ceZrcCommandsSupported](#) structure.

Parameters

<code>tag</code>	A pointer to an EmberAfRf4ceZrcCommandsSupported structure.
------------------	---

Returns

Returns a pointer to the first byte of the command discovery value.

6.10 ZigBee Remote Control 2.0 Profile

Macros

- #define ACTION_MAPPING_CLIENT
- #define ACTION_MAPPING_SERVER
- #define SET_DEFAULT(entry)

Functions

- EmberStatus emberAfRf4ceZrc20Bind (EmberAfRf4ceDeviceType searchDevType)
- EmberStatus emberAfRf4ceZrc20ProxyBind (EmberPanId panId, EmberEUI64 ieeeAddr)
- EmberStatus emberAfRf4ceZrc20ActionStart (uint8_t pairingIndex, EmberAfRf4ceZrcActionBank actionBank, EmberAfRf4ceZrcActionCode actionCode, EmberAfRf4ceZrcModifierBit actionModifier, uint16_t actionVendorId, const uint8_t *actionData, uint8_t actionDataLength, bool atomic)
- EmberStatus emberAfRf4ceZrc20ActionStop (uint8_t pairingIndex, EmberAfRf4ceZrcActionBank actionBank, EmberAfRf4ceZrcActionCode actionCode, EmberAfRf4ceZrcModifierBit actionModifier, uint16_t actionVendorId)
- EmberStatus emberAfRf4ceZrc20LegacyCommandDiscovery (uint8_t pairingIndex)
- EmberStatus emberAfRf4ceZrc20StartActionMappingsNegotiation (uint8_t pairingIndex)
- EmberStatus emberAfRf4ceZrc20StartHomeAutomationSupportedAnnouncement (uint8_t pairingIndex)
- EmberStatus emberAfRf4ceZrc20PullHomeAutomationAttribute (uint8_t pairingIndex, uint16_t vendorId, uint8_t haInstanceId, uint8_t haAttributeId)
- uint8_t * emberAfRf4ceZrcCommandsSupportedContents (EmberAfRf4ceZrcCommandsSupported *commandsSupported)
- bool emberAfRf4ceZrc20ActionMappingEntryHasRfDescriptor (const EmberAfRf4ceZrcActionMapping *entry)
- bool emberAfRf4ceZrc20ActionMappingEntryHasIrDescriptor (const EmberAfRf4ceZrcActionMapping *entry)
- bool emberAfRf4ceZrc20ActionMappingEntryHasIrVendorId (const EmberAfRf4ceZrcActionMapping *entry)

6.10.1 Detailed Description

The ZigBee Remote Control 2.0 (ZRC2.0) plugin implements the ZRC2.0 profile. Compared to versions 1.0 and 1.1 of the profile, ZRC2.0 describes a more complex discovery, pairing, configuration, and validation procedure for binding devices and a more robust mechanism for controlling devices via action commands. The ZRC2.0 profile is based on the Generic Device Profile (GDP). In conjunction with the [Generic Device Profile](#) plugin, this plugin manages these procedures for both originators and recipients. Note that this plugin implements version 2.0 of the ZRC profile. Version 2.0 devices are fully interoperable with version 1.0 and 1.1 devices. (ZRC1.0 was also known as the Consumer Electronics Remote Control (CERC) profile.) As required by the specification, this plugin sends ZRC1.1-formatted messages to ZRC1.x devices and ZRC2.0-formatted messages to ZRC2.0 devices. It receives ZRC1.0-, ZRC1.1-, and ZRC2.0-formatted messages.

This plugin supports originators and recipients. Originators send action messages to recipients after binding with them. Originators are most typically controllers but may also be targets. Recipients must be targets. Both originators and recipients must start general network operations before beginning ZRC-specific operations. Network operations should be started by calling [emberAfRf4ceStart](#) in the [RF4CE Profile Support](#) plugin.

Once network operations have started, ZRC2.0 originators can initiate the binding procedure by calling `emberAfRf4ceZrc20Bind` with the requested search device type. This causes the GDP plugin to perform discovery for matching targets in range. Potential targets are ranked according to an algorithm described in the GDP specification. If one or more potential targets are identified, the GDP plugin will attempt to pair with the highest-ranked target. Once the temporary pairing is established, both the GDP and ZRC profiles will perform their respective configuration procedures. Finally, the validation procedure begins. Validation is implementation specific, but may be as simple as a button press on the target or a more involved challenge-response mechanism between target and originator. Only if validation is successful are the originator and recipient considered bound. If any step fails, the GDP plugin will restart the binding procedure using the target that has the next-highest rank. At the conclusion of the binding procedure, the GDP plugin will call `emberAfPluginRf4ceGdpBindingCompleteCallback` to indicate whether binding completed successfully with a target and, if so, which pairing index has been assigned to that target.

The GDP plugin and this plugin will manage discovery, pairing, and configuration for recipients on behalf of the device. If configuration completes successfully, the GDP plugin will call `emberAfPluginRf4ceGdpStartValidationCallback` so that the device can begin the implementation-specific validation procedure. Once the application determines the validation status of the originator, it should call `emberAfRf4ceGdpSetValidationStatus`. Alternatively, the recipient may call `emberAfRf4ceGdpPushButton` to set the push-button stimulus flag. Once set, the GDP plugin will automatically validate binding attempts. Push-button mode lasts for a fixed duration as described in the GDP specification and may be initiated at any time before or during binding, including during the validation procedure.

Following a successful binding, action originators may send action messages to action recipients by calling `emberAfRf4ceZrc20ActionStart`. In response, the plugin will transmit a start or atomic message to the indicated target with specific action parameters. If the action is atomic, the plugin performs no additional processing. Otherwise, the plugin will repeatedly transmit repeat messages at fixed intervals until `emberAfRf4ceZrc20ActionStop` is called. The interval at which repeat messages are transmitted is configurable in the plugin options.

For action recipients, the plugin will keep track of incoming action messages. Each time a start or atomic command is received, the plugin will call `emberAfPluginRf4ceZrc20ActionCallback` with its information. For non-atomic actions, the plugin will then wait for a fixed duration for a corresponding repeat action. If a repeat action is received within the timeout, the plugin will reset its timer and wait for the next message. If a repeat message is not received within the timeout, the plugin will call `emberAfPluginRf4ceZrc20ActionCallback` with an indication that the action has stopped. The timeout for receiving repeat messages is configurable in the plugin options. Note that the plugin will not call `emberAfPluginRf4ceZrc20ActionCallback` for repeat messages that follow a start message.

The plugin is capable of keeping track of a fixed number of simultaneous incoming and outgoing actions. The limits are configurable in the plugin options.

This plugin manages the state of the receiver by calling `emberAfRf4ceRxEnable` using `EMBER_AF_RF4CE_PROFILE_REMOTE_CONTROL_2_0` as the profile id. If the application also wishes to manage the receiver, it should do so using `EMBER_AF_RF4CE_PROFILE_WILDCARD` as the profile id or by calling `emberAfRf4ceSetPowerSavingParameters`.

This plugin utilizes the discovery, pairing, sending and receiving, and power-saving functionality provided by the [RF4CE Profile Support](#) plugin. It also utilizes the binding functionality provided by the [Generic Device Profile](#) plugin. Support for the optional Action Mapping feature in ZRC2.0 is provided by the [ZigBee Remote Control 2.0 Action Mapping Client](#) and [ZigBee Remote Control 2.0 Action Mapping Server](#) plugins. Support for the optional Home Automation (HA) interoperability feature is provided by the [ZigBee Remote Control 2.0 Home Automation Client](#) and [ZigBee Remote Control 2.0 Home Automation Server](#) plugins.

6.10.2 Macro Definition Documentation

6.10.2.1 #define ACTION_MAPPING_CLIENT

Definition at line 117 of file [rf4ce-zrc20.h](#).

6.10.2.2 #define ACTION_MAPPING_SERVER

Definition at line 118 of file [rf4ce-zrc20.h](#).

6.10.2.3 #define SET_DEFAULT(entry)

Definition at line 343 of file [rf4ce-zrc20.h](#).

6.10.3 Function Documentation

6.10.3.1 EmberStatus emberAfRf4ceZrc20Bind (EmberAfRf4ceDeviceType *searchDevType*)

6.10.3.2 EmberStatus emberAfRf4ceZrc20ProxyBind (EmberPanId *panId*, EmberEUI64 *ieeeAddr*)

6.10.3.3 EmberStatus emberAfRf4ceZrc20ActionStart (uint8_t *pairingIndex*, EmberAfRf4ceZrcActionBank *actionBank*, EmberAfRf4ceZrcActionCode *actionCode*, EmberAfRf4ceZrcModifierBit *actionModifier*, uint16_t *actionVendorId*, const uint8_t * *actionData*, uint8_t *actionDataLength*, bool *atomic*)

Adds the passed action record to the internal list of outstanding action records. If the passed action record is already in the internal record list, the action type for the matching record is set back to 'start' and a new actions command is sent out right away. This API ensures that the passed record fits in the next outgoing actions command.

Parameters

<i>pairingIndex</i>	The index of the pairing the passed action record is destined to.
<i>actionBank</i>	The action bank field of the action record.
<i>actionCode</i>	The action code field of the action record.
<i>actionModifier</i>	The modifier bits of the action record.
<i>actionVendorId</i>	The vendor ID to be included in the action record. A value of ::EMBER_RF4CE_NULL_VENDOR_ID won't be included in the over-the-air action record.
<i>actionData</i>	A pointer to the <i>actionData</i> field of the action record. Notice that this memory area is not copied, therefore it should refer some global memory area. A NULL pointer causes this field to not be included in the over-the-air action record.
<i>actionDataLength</i>	The length in bytes of the <i>actionData</i> field.

Returns

An ::EmberStatus value of ::EMBER_SUCCESS if there is a free entry in the table of outstanding action records and the passed action record fits in the next outgoing actions command for the passed pairing index. Otherwise it returns an ::EmberStatus value of ::EMBER_INVALID_CALL.

6.10.3.4 EmberStatus emberAfRf4ceZrc20ActionStop (`uint8_t pairingIndex`, `EmberAfRf4ceZrcActionBank actionBank`, `EmberAfRf4ceZrcActionCode actionCode`, `EmberAfRf4ceZrcModifierBit actionModifier`, `uint16_t actionVendorId`)

Removes the passed action record from the internal list of outstanding action records.

Parameters

<code>pairingIndex</code>	The index of the pairing the passed action record is destined to.
<code>actionBank</code>	The action bank field of the action record to be removed.
<code>actionCode</code>	The action code field of the action record to be removed.
<code>actionModifier</code>	The modifier bits of the action record to be removed.

Returns

An ::EmberStatus value of ::EMBER_SUCCESS if the passed action record was successfully found and removed from the table of outstanding action records. Otherwise it returns an ::EmberStatus value of ::EMBER_INVALID_CALL.

6.10.3.5 EmberStatus emberAfRf4ceZrc20LegacyCommandDiscovery (`uint8_t pairingIndex`)

Initiates the legacy ZRC 1.1 command discovery process. If this API returns a successful status, the corresponding emberAfRf4ceZrc20LegacyCommandDiscoveryComplete() callback will be called upon receiving the discovery response command from the peer node or upon timeout.

Parameters

<code>pairingIndex</code>	The index of the destination pairing.
---------------------------	---------------------------------------

Returns

An ::EmberStatus value of ::EMBER_SUCCESS if a ZRC 1.1 Discovery Request command was sent to the pairing corresponding to the passed pairing index. An ::EmberStatus value of ::EMBER_INVALID_CALL if the destination pairing supports ZRC 2.0 or does not support ZRC 1.1. Otherwise it returns an ::EmberStatus value indicating the TX failure reason.

6.10.3.6 EmberStatus emberAfRf4ceZrc20StartActionMappingsNegotiation (`uint8_t pairingIndex`)

Initiates the action mapping negotiation procedure at the action mapping client with the action mapping server.

Parameters

<code>pairingIndex</code>	The index of the action mapping server pairing.
---------------------------	---

Returns

An ::EmberStatus value of ::EMBER_SUCCESS if the action mapping negotiation procedure is successfully initiated. Otherwise it returns an ::EmberStatus value of ::EMBER_INVALID_CALL.

6.10.3.7 EmberStatus emberAfRf4ceZrc20StartHomeAutomationSupportedAnnouncement (uint8_t pairingIndex)

Initiates the Home Automation supported announcement procedure at the Home Automation originator with the Home Automation recipient.

Parameters

<i>pairingIndex</i>	The index of the Home Automation recipient pairing.
---------------------	---

Returns

An ::EmberStatus value of ::EMBER_SUCCESS if the Home Automation supported announcement procedure is successfully initiated. Otherwise it returns an ::EmberStatus value of ::EMBER_INVALID_CALL.

6.10.3.8 EmberStatus emberAfRf4ceZrc20PullHomeAutomationAttribute (uint8_t pairingIndex, uint16_t vendorId, uint8_t haInstanceId, uint8_t haAttributelD)

Pulls a Home Automation attribute from the Home Automation recipient.

Parameters

<i>pairingIndex</i>	The index of the Home Automation recipient pairing.
<i>vendorId</i>	The vendor ID to be included in the Pull command. If a value of ::EMBER_RF4CE_NULL_VENDOR_ID is passed, no vendor ID will be included in the Pull command.
<i>haInstanceId</i>	The Home Automation instance ID.
<i>haAttributelD</i>	The Home Automation attribute ID.

Returns

An ::EmberStatus value of ::EMBER_SUCCESS if the Home Automation Pull command was successfully sent. If this is the case, the corresponding callback ::emberAfPluginRf4ceZrc20PullHomeAutomationAttributeCompleteCallback() will fire. Otherwise it returns an ::EmberStatus indicating the reason of failure.

6.10.3.9 uint8_t* emberAfRf4ceZrcCommandsSupportedContents (EmberAfRf4ceZrcCommandsSupported * commandsSupported)

Accesses the actual command discovery data of the [EmberAfRf4ceZrcCommandsSupported](#) structure.

Parameters

<i>tag</i>	A pointer to an EmberAfRf4ceZrcCommandsSupported structure.
------------	---

Returns

uint8_t* Returns a pointer to the first byte of the command discovery value.

6.10.3.10 bool emberAfRf4ceZrc20ActionMappingEntryHasRfDescriptor (const EmberAfRf4ceZrcActionMapping * entry)

Determine if the action mapping entry has an RF descriptor.

Implemented as a macro for efficiency.

Returns

true if the action mapping table entry has an RF descriptor or false otherwise.

6.10.3.11 bool emberAfRf4ceZrc20ActionMappingEntryHasIrDescriptor (const EmberAfRf4ceZrcActionMapping * entry)

Determine if the action mapping entry has an IR descriptor.

Implemented as a macro for efficiency.

Returns

true if the action mapping table entry has an IR descriptor or false otherwise.

6.10.3.12 bool emberAfRf4ceZrc20ActionMappingEntryHasIrVendorId (const EmberAfRf4ceZrcActionMapping * entry)

Determine if the action mapping entry has an IR vendor ID.

Implemented as a macro for efficiency.

Returns

true if the action mapping table entry has an IR vendor ID or false otherwise.

6.11 ZigBee Remote Control 2.0 Action Mapping Client

Functions

- void [emberAfRf4ceZrc20ActionMappingClientClearAllActionMappings](#) (void)
- EmberStatus [emberAfRf4ceZrc20ActionMappingClientClearActionMappingsPerPairing](#) (uint8_t pairingIndex)
- EmberStatus [emberAfRf4ceZrc20ActionMappingClientClearActionMapping](#) (uint8_t pairingIndex, EmberAfRf4ceDeviceType actionDeviceType, EmberAfRf4ceZrcActionBank actionBank, EmberAfRf4ceZrcActionCode actionCode)
- EmberStatus [emberAfRf4ceZrc20ActionMappingClientGetActionMapping](#) (uint8_t pairingIndex, EmberAfRf4ceDeviceType actionDeviceType, EmberAfRf4ceZrcActionBank actionBank, EmberAfRf4ceZrcActionCode actionCode, EmberAfRf4ceZrcActionMapping *actionMapping)
- EmberStatus [emberAfRf4ceZrc20ActionMappingClientLookUpActionMapping](#) (uint8_t pairingIndex, EmberAfRf4ceDeviceType actionDeviceType, EmberAfRf4ceZrcActionBank actionBank, EmberAfRf4ceZrcActionCode actionCode, EmberAfRf4ceZrcActionMapping *actionMapping)
- EmberStatus [emberAfRf4ceZrc20ActionMappingClientSetActionMapping](#) (uint8_t pairingIndex, EmberAfRf4ceDeviceType actionDeviceType, EmberAfRf4ceZrcActionBank actionBank, EmberAfRf4ceZrcActionCode actionCode, EmberAfRf4ceZrcActionMapping *actionMapping)

6.11.1 Detailed Description

The ZigBee Remote Control 2.0 (ZRC2.0) Action Mapping Client plugin implements the optional action mapping feature of the ZRC2.0 profile for clients. Action mapping provides a standard mechanism for remapping actions and is generally used to allow a remote to control legacy IR devices or to perform simultaneous IR and RF functions. This plugin manages the storage and retrieval of these mappings for action originators.

This plugin provides information to the [ZigBee Remote Control 2.0 Profile](#) plugin about which mappable actions this device supports. A mappable action is simply an action on the local device that may be remapped by the server to an IR code, some other RF action, or to some IR-RF combination. During binding, the mappable actions supported by this device are exchanged with the server. The server uses this information to set up the appropriate remappings. The mappable actions are configurable in AppBuilder.

When the server provides action mappings, either during the binding procedure or when the application manually queries the server for them, the [ZigBee Remote Control 2.0 Profile](#) plugin receives the mappings and passes them to this plugin via [emberAfPluginRf4ceZrc20IncomingActionMappingCallback](#). This plugin, in turn, will store the data in RAM. The amount of storage space dedicated to storing the mappings is configurable via the plugin options.

Prior to starting an action, the application should call [emberAfRf4ceZrc20ActionMappingClientGetActionMapping](#) to determine if the action has been remapped. If the action has been remapped and an IR code has been specified, the application should transmit it using application-specific means. If an RF mapping has been specified, the application should pass it to [emberAfRf4ceZrc20ActionStart](#) instead of the original action.

Action mappings are generally managed by the server, but the application is provided some control as well. Specific mappings can be manipulated by calling [emberAfRf4ceZrc20ActionMappingClientSetActionMapping](#). An action may be set to the default behavior by calling [emberAfRf4ceZrc20ActionMappingClientClearActionMapping](#). Resetting all actions to the defaults can be accomplished on a per-pairing basis by calling [emberAfRf4ceZrc20ActionMappingClientClearActionMappingsPerPairing](#) or for all pairings by calling [emberAfRf4ceZrc20ActionMappingClientClearAllActionMappings](#). It may be desirable to reset action mappings when the device unpairs from a specific server or is reset to factory new.

Support for the optional action mapping feature for servers is provided by the [ZigBee Remote Control 2.0 Action Mapping Server](#) plugin.

6.11.2 Function Documentation

6.11.2.1 void emberAfRf4ceZrc20ActionMappingClientClearAllActionMappings (void)

Clear all action mappings on the client.

6.11.2.2 EmberStatus emberAfRf4ceZrc20ActionMappingClientClearActionMappingsPerPairing (uint8_t pairingIndex)

Clear action mappings per pairing on the client.

Parameters

<i>pairingIndex</i>	The pairing index the clear action mapping command should be sent to.
---------------------	---

Returns

An ::EmberStatus value indicating whether the clear action mappings per pairing command was successfully sent out or the reason of failure.

6.11.2.3 EmberStatus emberAfRf4ceZrc20ActionMappingClientClearActionMapping (uint8_t pairingIndex, EmberAfRf4ceDeviceType *actionDeviceType*, EmberAfRf4ceZrcActionBank *actionBank*, EmberAfRf4ceZrcActionCode *actionCode*)

Clear action mapping that belongs to a mappable action.

Parameters

<i>pairingIndex</i>	The pairing index the clear action mapping command should be sent to.
<i>actionDevice-Type</i>	The action device type of the mappable action.
<i>actionBank</i>	The action bank of the mappable action.
<i>actionCode</i>	The action code of the mappable action.

Returns

An ::EmberStatus value indicating whether the clear action mapping command was successfully sent out or the reason of failure.

6.11.2.4 EmberStatus emberAfRf4ceZrc20ActionMappingClientGetActionMapping (uint8_t pairingIndex, EmberAfRf4ceDeviceType *actionDeviceType*, EmberAfRf4ceZrcActionBank *actionBank*, EmberAfRf4ceZrcActionCode *actionCode*, EmberAfRf4ceZrcActionMapping * *actionMapping*)

Get action mapping corresponding to a mappable action.

Parameters

<i>pairingIndex</i>	The pairing index the get action mapping command should be sent to.
<i>actionDevice-Type</i>	The action device type of the mappable action.
<i>actionBank</i>	The action bank of the mappable action.
<i>actionCode</i>	The action code of the mappable action.
<i>actionMapping</i>	The action mapping structure describing the action mapping.

Returns

An ::EmberStatus value indicating whether the get action mapping command was successfully sent out or the reason of failure.

6.11.2.5 EmberStatus emberAfRf4ceZrc20ActionMappingClientLookUpActionMapping (uint8_t pairingIndex, EmberAfRf4ceDeviceType *actionDeviceType*, EmberAfRf4ceZrcActionBank *actionBank*, EmberAfRf4ceZrcActionCode *actionCode*, EmberAfRf4ceZrcActionMapping * *actionMapping*)

Look up action mapping that belongs to a mappable action.

Parameters

<i>pairingIndex</i>	The pairing index the look up action mapping command should be sent to.
<i>actionDevice-Type</i>	The action device type of the mappable action.
<i>actionBank</i>	The action bank of the mappable action.
<i>actionCode</i>	The action code of the mappable action.
<i>actionMapping</i>	The action mapping structure describing the action mapping.

Returns

An ::EmberStatus value indicating whether the look up action mapping command was successfully sent out or the reason of failure.

Deprecated This function is deprecated and will be removed in a future release. Customers should use [emberAfRf4ceZrc20ActionMappingClientGetActionMapping](#) instead.

6.11.2.6 EmberStatus emberAfRf4ceZrc20ActionMappingClientSetActionMapping (uint8_t *pairingIndex*, EmberAfRf4ceDeviceType *actionDeviceType*, EmberAfRf4ceZrcActionBank *actionBank*, EmberAfRf4ceZrcActionCode *actionCode*, EmberAfRf4ceZrcActionMapping * *actionMapping*)

Set action mapping corresponding to a mappable action.

Parameters

<i>pairingIndex</i>	The pairing index the set action mapping command should be sent to.
<i>actionDevice-Type</i>	The action device type of the mappable action.
<i>actionBank</i>	The action bank of the mappable action.
<i>actionCode</i>	The action code of the mappable action.
<i>actionMapping</i>	The action mapping structure describing the action mapping.

Returns

An ::EmberStatus value indicating whether the set action mapping command was successfully sent out or the reason of failure.

6.12 ZigBee Remote Control 2.0 Action Mapping Server

Functions

- EmberStatus [emberAfRf4ceZrc20ActionMappingServerRemapAction](#) (*EmberAfRf4ceZrcMappableAction *mappableAction, EmberAfRf4ceZrcActionMapping *actionMapping*)
- EmberStatus [emberAfRf4ceZrc20ActionMappingServerRestoreDefaultAction](#) (*EmberAfRf4ceZrcMappableAction *mappableAction*)
- void [emberAfRf4ceZrc20ActionMappingServerRestoreDefaultAllActions](#) (*void*)
- uint16_t [emberAfRf4ceZrc20ActionMappingServerGetMappableActionCount](#) (*void*)

6.12.1 Detailed Description

The ZigBee Remote Control 2.0 (ZRC2.0) Action Mapping Server plugin implements the optional action mapping feature of the ZRC2.0 profile for servers. Action mapping provides a standard mechanism for remapping actions and is generally used to allow a remote to control legacy IR devices or to perform simultaneous IR and RF functions. This plugin manages the storage and retrieval of these mappings for action recipients.

This plugin provides information to the [ZigBee Remote Control 2.0 Profile](#) plugin about action mappings for clients. An action mapping remaps an action on the client to an IR code, some other RF action, or to some IR-RF combination. During binding, the mappable actions supported by the client are exchanged with this device. The [ZigBee Remote Control 2.0 Profile](#) plugin receives the actions and passes them to this plugin via [emberAfPluginRf4ceZrc20IncomingMappableActionCallback](#). This plugin, in turn, will store the data in RAM. The amount of storage space dedicated to storing mappable actions and action mappings is configurable via the plugin options.

To create a mapping, the application should call [emberAfRf4ceZrc20ActionMappingServerRemapAction](#) with the action to remap and the new mapping. Once created, the mapping will apply to all clients for that particular action. Clients will be notified of action mappings during the binding procedure or if they manually query the server for them after binding. How the remappings are determined is application specific, but may involve an interactive menu that a user must navigate to identify the appropriate mappings. If a mapping no longer applies, perhaps because a connected peripheral such as a television has changed, the mapping may be removed and the action restored to its default by calling [emberAfRf4ceZrc20ActionMappingServerRestoreDefaultAction](#). To reset all actions to the default, for example due to a reset to factory new, [emberAfRf4ceZrc20ActionMappingServerRestoreDefaultAllActions](#) may be called.

Support for the optional action mapping feature for clients is provided by the [ZigBee Remote Control 2.0 Action Mapping Client](#) plugin.

6.12.2 Function Documentation

6.12.2.1 EmberStatus [emberAfRf4ceZrc20ActionMappingServerRemapAction](#) (*EmberAfRf4ceZrcMappableAction * mappableAction, EmberAfRf4ceZrcActionMapping * actionMapping*)

Remap mappable action on the server.

Parameters

<i>mappableAction</i>	The mappable action structure describing the mappable action to remap.
<i>actionMapping</i>	The action mapping structure describing the action mapping to be written into the action mapping table.

Returns

An ::EmberStatus value indicating whether the remap action mapping command was successfully sent out or the reason of failure.

6.12.2.2 EmberStatus emberAfRf4ceZrc20ActionMappingServerRestoreDefaultAction (EmberAfRf4ceZrcMappableAction * *mappableAction*)

Restore default action of the mappable action on the server.

Parameters

<i>mappable-Action</i>	The mappable action structure describing the mappable action which action is restored to default.
------------------------	---

Returns

An ::EmberStatus value indicating whether the restore default action command was successfully sent out or the reason of failure.

6.12.2.3 void emberAfRf4ceZrc20ActionMappingServerRestoreDefaultAllActions (void)

Restore all actions to default on the server.

6.12.2.4 uint16_t emberAfRf4ceZrc20ActionMappingServerGetMappableActionCount (void)

Get the number of mappable actions on the server.

Returns

Mappable Action count

6.13 ZigBee Remote Control 2.0 Home Automation Client

Macros

- #define `emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene`(scene)
- #define `emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene`(scene)
- #define `emberAfRf4ceZrcHaFillCommandPreviousDestinationGroup()`
- #define `emberAfRf4ceZrcHaFillCommandNextDestinationGroup()`

Enumerations

- enum `EmberAfThermostatSetpointRaiseLowerMode` { EMBER_ZCL_THERMOSTAT_SETPOINT_RAISE_LOWER_MODE_HEAT, EMBER_ZCL_THERMOSTAT_SETPOINT_RAISE_LOWER_MODE_COOL, EMBER_ZCL_THERMOSTAT_SETPOINT_RAISE_LOWER_MODE_BOOTH }

Functions

- uint16_t `emAfPluginRf4ceZrc20HaFillExternalBuffer` (PGM_P format,...)
- EmberStatus `emberAfRf4ceZrcHaSend` (uint8_t pairingIndex, uint8_t haInstanceId)

Scenes Commands

- #define `emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene0()`
- #define `emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene1()`
- #define `emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene2()`
- #define `emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene3()`
- #define `emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene4()`
- #define `emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene5()`
- #define `emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene6()`
- #define `emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene7()`
- #define `emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene8()`
- #define `emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene9()`
- #define `emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene10()`
- #define `emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene11()`
- #define `emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene12()`
- #define `emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene13()`
- #define `emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene14()`
- #define `emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene15()`
- #define `emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene0()`
- #define `emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene1()`
- #define `emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene2()`
- #define `emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene3()`
- #define `emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene4()`
- #define `emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene5()`
- #define `emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene6()`
- #define `emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene7()`
- #define `emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene8()`
- #define `emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene9()`

- #define `emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene10()`
- #define `emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene11()`
- #define `emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene12()`
- #define `emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene13()`
- #define `emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene14()`
- #define `emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene15()`

On/off Commands

- #define `emberAfRf4ceZrcHaFillCommandOnOffClusterOff()`
- #define `emberAfRf4ceZrcHaFillCommandOnOffClusterOn()`
- #define `emberAfRf4ceZrcHaFillCommandOnOffClusterToggle()`

Level Control Commands

- #define `emberAfRf4ceZrcHaFillCommandLevelControlClusterMoveToLevel(level,transitionTime)`
- #define `emberAfRf4ceZrcHaFillCommandLevelControlClusterMove(moveMode,rate)`
- #define `emberAfRf4ceZrcHaFillCommandLevelControlClusterStep(stepMode,stepSize,transitionTime)`
- #define `emberAfRf4ceZrcHaFillCommandLevelControlClusterStop()`
- #define `emberAfRf4ceZrcHaFillCommandLevelControlClusterMoveToLevelWithOnOff(level,transitionTime)`
- #define `emberAfRf4ceZrcHaFillCommandLevelControlClusterMoveWithOnOff(moveMode, rate)`
- #define `emberAfRf4ceZrcHaFillCommandLevelControlClusterStepWithOnOff(stepMode,stepSize,transitionTime)`
- #define `emberAfRf4ceZrcHaFillCommandLevelControlClusterStopWithOnOff()`

Door Lock Commands

- #define `emberAfRf4ceZrcHaFillCommandDoorLockClusterLockDoor(pinRfidCode)`
- #define `emberAfRf4ceZrcHaFillCommandDoorLockClusterUnlockDoor(pinRfidCode)`
- #define `emberAfRf4ceZrcHaFillCommandDoorLockClusterToggle(pinRfidCode)`
- #define `emberAfRf4ceZrcHaFillCommandDoorLockClusterUnlockWithTimeout(timeoutInSeconds,pinRfidCode)`

Window Covering Commands

- #define `emberAfRf4ceZrcHaFillCommandWindowCoveringClusterUpOpen()`
- #define `emberAfRf4ceZrcHaFillCommandWindowCoveringClusterDownClose()`
- #define `emberAfRf4ceZrcHaFillCommandWindowCoveringClusterStop()`
- #define `emberAfRf4ceZrcHaFillCommandWindowCoveringClusterGoToLiftValue(liftValue)`
- #define `emberAfRf4ceZrcHaFillCommandWindowCoveringClusterGoToLiftPercentage(percentageLiftValue)`
- #define `emberAfRf4ceZrcHaFillCommandWindowCoveringClusterGoToTiltValue(tiltValue)`
- #define `emberAfRf4ceZrcHaFillCommandWindowCoveringClusterGoToTiltPercentage(percentageTiltValue)`

Thermostat Commands

- #define `emberAfRf4ceZrcHaFillCommandThermostatClusterSetpointRaiseLower(mode,amount)`

Color Control Commands

- #define `emberAfRf4ceZrcHaFillCommandColorControlClusterMoveToHue`(hue,direction,transition-Time)
- #define `emberAfRf4ceZrcHaFillCommandColorControlClusterMoveHue`(moveMode,rate)
- #define `emberAfRf4ceZrcHaFillCommandColorControlClusterStepHue`(stepMode,stepSize,transition-Time)
- #define `emberAfRf4ceZrcHaFillCommandColorControlClusterMoveToSaturation`(saturation,transition-Time)
- #define `emberAfRf4ceZrcHaFillCommandColorControlClusterMoveSaturation`(moveMode, rate)
- #define `emberAfRf4ceZrcHaFillCommandColorControlClusterStepSaturation`(stepMode,stepSize,transition-Time)
- #define `emberAfRf4ceZrcHaFillCommandColorControlClusterMoveToHueAndSaturation`(hue,saturation,transition-Time)
- #define `emberAfRf4ceZrcHaFillCommandColorControlClusterMoveToColor`(colorX,colorY,transition-Time)
- #define `emberAfRf4ceZrcHaFillCommandColorControlClusterMoveColor`(rateX,rateY)
- #define `emberAfRf4ceZrcHaFillCommandColorControlClusterStepColor`(stepX,stepY,transitionTime)
- #define `emberAfRf4ceZrcHaFillCommandColorControlClusterMoveToColorTemperature`(colorTemperature, transitionTime)

IAS ACE Commands

- #define `emberAfRf4ceZrcHaFillCommandIASACEClusterArm`(armMode)
- #define `emberAfRf4ceZrcHaFillCommandIASACEClusterBypass`(numberOfZones,zoneIds,zoneIds-Len)
- #define `emberAfRf4ceZrcHaFillCommandIASACEClusterEmergency`()
- #define `emberAfRf4ceZrcHaFillCommandIASACEClusterFire`()
- #define `emberAfRf4ceZrcHaFillCommandIASACEClusterPanic`()

6.13.1 Detailed Description

The ZigBee Remote Control 2.0 (ZRC2.0) Home Automation (HA) Client plugin implements the optional HA interoperability feature of the ZRC2.0 profile for HA action originators. HA interoperability allows RF4CE devices to control HA devices operating on a ZigBee PRO network within the same premises.

HA action originators are most typically controllers but may also be targets. HA action recipients must be targets. HA originators transmit HA commands to an HA recipient over the RF4CE network as ZR-C2.0 actions. The recipient then translates these actions to ZigBee Cluster Library (ZCL) messages and retransmits them to the appropriate HA device on the PRO network. The originator directs all actions to abstract HA instances. Each HA action recipient is responsible for mapping HA instances to actual physical devices.

After binding with an HA action recipient, the originator can construct an HA command by calling one of the `emberAfZrcHaFill` APIs. The message can then be sent to the recipient by calling `emberAfRf4ce-ZrcHaSend`. For example, the application can toggle the light bulb identified as HA instance 0 on its first pairing by calling `emberAfRf4ceZrcHaFillCommandOnOffClusterToggle` followed by `::emberAfRf4ce-ZrcHaSend(0, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_0)`.

This plugin also provides information to the [ZigBee Remote Control 2.0 Profile](#) plugin about which HA attributes this device is interested in. During binding, this information is exchanged with the HA action recipient. The recipient will use this information to send notifications when those attributes change on the HA network. The supported attributes are configurable in AppBuilder.

This plugin utilizes the action functionality provided by the [ZigBee Remote Control 2.0 Profile](#) plugin. Support for the optional HA action recipient feature is provided by the [ZigBee Remote Control 2.0 Home Automation Server](#) plugin.

6.13.2 Macro Definition Documentation

6.13.2.1 `#define emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene(scene)`

Definition at line [53](#) of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.2 `#define emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene(scene)`

Definition at line [57](#) of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.3 `#define emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene0()`

Store local scene 0.

Cluster: Scenes, Provides an interface for storing local scene 0. Command: StoreLocalScene0

Definition at line [87](#) of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.4 `#define emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene1()`

Store local scene 1.

Cluster: Scenes, Provides an interface for storing local scene 1. Command: StoreLocalScene1

Definition at line [96](#) of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.5 `#define emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene2()`

Store local scene 2.

Cluster: Scenes, Provides an interface for storing local scene 2. Command: StoreLocalScene2

Definition at line [105](#) of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.6 `#define emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene3()`

Store local scene 3.

Cluster: Scenes, Provides an interface for storing local scene 3. Command: StoreLocalScene3

Definition at line [114](#) of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.7 `#define emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene4()`

Store local scene 4.

Cluster: Scenes, Provides an interface for storing local scene 4. Command: StoreLocalScene4

Definition at line [123](#) of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.8 #define emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene5()

Store local scene 5.

Cluster: Scenes, Provides an interface for storing local scene 5. Command: StoreLocalScene5

Definition at line 132 of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.9 #define emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene6()

Store local scene 6.

Cluster: Scenes, Provides an interface for storing local scene 6. Command: StoreLocalScene6

Definition at line 141 of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.10 #define emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene7()

Store local scene 7.

Cluster: Scenes, Provides an interface for storing local scene 7. Command: StoreLocalScene7

Definition at line 150 of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.11 #define emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene8()

Store local scene 8.

Cluster: Scenes, Provides an interface for storing local scene 8. Command: StoreLocalScene8

Definition at line 159 of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.12 #define emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene9()

Store local scene 9.

Cluster: Scenes, Provides an interface for storing local scene 9. Command: StoreLocalScene9

Definition at line 168 of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.13 #define emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene10()

Store local scene 10.

Cluster: Scenes, Provides an interface for storing local scene 10. Command: StoreLocalScene10

Definition at line 177 of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.14 #define emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene11()

Store local scene 11.

Cluster: Scenes, Provides an interface for storing local scene 11. Command: StoreLocalScene11

Definition at line 186 of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.15 #define emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene12()

Store local scene 12.

Cluster: Scenes, Provides an interface for storing local scene 12. Command: StoreLocalScene12

Definition at line 195 of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.16 #define emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene13()

Store local scene 13.

Cluster: Scenes, Provides an interface for storing local scene 13. Command: StoreLocalScene13

Definition at line 204 of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.17 #define emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene14()

Store local scene 14.

Cluster: Scenes, Provides an interface for storing local scene 14. Command: StoreLocalScene14

Definition at line 213 of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.18 #define emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene15()

Store local scene 15.

Cluster: Scenes, Provides an interface for storing local scene 15. Command: StoreLocalScene15

Definition at line 222 of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.19 #define emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene0()

Recall local scene 0.

Cluster: Scenes, Provides an interface for recalling local scene 0. Command: RecallLocalScene0

Definition at line 231 of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.20 #define emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene1()

Recall local scene 1.

Cluster: Scenes, Provides an interface for recalling local scene 1. Command: RecallLocalScene1

Definition at line 240 of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.21 #define emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene2()

Recall local scene 2.

Cluster: Scenes, Provides an interface for recalling local scene 2. Command: RecallLocalScene2

Definition at line 249 of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.22 #define emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene3()

Recall local scene 3.

Cluster: Scenes, Provides an interface for recalling local scene 3. Command: RecallLocalScene3

Definition at line [258](#) of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.23 #define emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene4()

Recall local scene 4.

Cluster: Scenes, Provides an interface for recalling local scene 4. Command: RecallLocalScene4

Definition at line [267](#) of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.24 #define emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene5()

Recall local scene 5.

Cluster: Scenes, Provides an interface for recalling local scene 5. Command: RecallLocalScene5

Definition at line [276](#) of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.25 #define emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene6()

Recall local scene 6.

Cluster: Scenes, Provides an interface for recalling local scene 6. Command: RecallLocalScene6

Definition at line [285](#) of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.26 #define emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene7()

Recall local scene 7.

Cluster: Scenes, Provides an interface for recalling local scene 7. Command: RecallLocalScene7

Definition at line [294](#) of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.27 #define emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene8()

Recall local scene 8.

Cluster: Scenes, Provides an interface for recalling local scene 8. Command: RecallLocalScene8

Definition at line [303](#) of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.28 #define emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene9()

Recall local scene 9.

Cluster: Scenes, Provides an interface for recalling local scene 9. Command: RecallLocalScene9

Definition at line [312](#) of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.29 #define emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene10()

Recall local scene 10.

Cluster: Scenes, Provides an interface for recalling local scene 10. Command: RecallLocalScene10

Definition at line [321](#) of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.30 #define emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene11()

Recall local scene 11.

Cluster: Scenes, Provides an interface for recalling local scene 11. Command: RecallLocalScene11

Definition at line [330](#) of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.31 #define emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene12()

Recall local scene 12.

Cluster: Scenes, Provides an interface for recalling local scene 12. Command: RecallLocalScene12

Definition at line [339](#) of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.32 #define emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene13()

Recall local scene 13.

Cluster: Scenes, Provides an interface for recalling local scene 13. Command: RecallLocalScene13

Definition at line [348](#) of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.33 #define emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene14()

Recall local scene 14.

Cluster: Scenes, Provides an interface for recalling local scene 14. Command: RecallLocalScene14

Definition at line [357](#) of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.34 #define emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene15()

Recall local scene 15.

Cluster: Scenes, Provides an interface for recalling local scene 15. Command: RecallLocalScene15

Definition at line [366](#) of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.35 #define emberAfRf4ceZrcHaFillCommandOnOffClusterOff()

Command description for Off.

Cluster: On/off, Attributes and commands for switching devices between 'On' and 'Off' states. Command: Off

Definition at line [378](#) of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.36 #define emberAfRf4ceZrcHaFillCommandOnOffClusterOn()

Command description for On.

Cluster: On/off, Attributes and commands for switching devices between 'On' and 'Off' states. Command: On

Definition at line 388 of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.37 #define emberAfRf4ceZrcHaFillCommandOnOffClusterToggle()

Command description for Toggle.

Cluster: On/off, Attributes and commands for switching devices between 'On' and 'Off' states. Command: Toggle

Definition at line 398 of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.38 #define emberAfRf4ceZrcHaFillCommandLevelControlClusterMoveToLevel(*level*, *transitionTime*)

Command description for Move To Level.

Cluster: Level Control, Attributes and commands for controlling devices that can be set to a level between fully 'On' and fully 'Off.' Command: MoveToLevel

Parameters

<i>level</i>	uint8_t
<i>transitionTime</i>	uint16_t

Definition at line 413 of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.39 #define emberAfRf4ceZrcHaFillCommandLevelControlClusterMove(*moveMode*, *rate*)

Command description for Move.

Cluster: Level Control, Attributes and commands for controlling devices that can be set to a level between fully 'On' and fully 'Off.' Command: Move

Parameters

<i>moveMode</i>	uint8_t
<i>rate</i>	uint8_t

Definition at line 428 of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.40 #define emberAfRf4ceZrcHaFillCommandLevelControlClusterStep(*stepMode*, *stepSize*, *transitionTime*)

Command description for Step.

Cluster: Level Control, Attributes and commands for controlling devices that can be set to a level between fully 'On' and fully 'Off.' Command: Step

Parameters

<i>stepMode</i>	uint8_t
<i>stepSize</i>	uint8_t
<i>transitionTime</i>	uint16_t

Definition at line 444 of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.41 #define emberAfRf4ceZrcHaFillCommandLevelControlClusterStop()

Command description for Stop.

Cluster: Level Control, Attributes and commands for controlling devices that can be set to a level between fully 'On' and fully 'Off.' Command: Stop

Definition at line 459 of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.42 #define emberAfRf4ceZrcHaFillCommandLevelControlClusterMoveToLevelWithOnOff(*level*, *transitionTime*)

Command description for Move To Level With On/Off.

Cluster: Level Control, Attributes and commands for controlling devices that can be set to a level between fully 'On' and fully 'Off.' Command: Move To Level With On/Off

Parameters

<i>level</i>	uint8_t
<i>transitionTime</i>	uint16_t

Definition at line 471 of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.43 #define emberAfRf4ceZrcHaFillCommandLevelControlClusterMoveWithOnOff(*moveMode*, *rate*)

Command description for Move With On/Off.

Cluster: Level Control, Attributes and commands for controlling devices that can be set to a level between fully 'On' and fully 'Off.' Command: Move With On/Off

Parameters

<i>moveMode</i>	uint8_t
<i>rate</i>	uint8_t

Definition at line 486 of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.44 #define emberAfRf4ceZrcHaFillCommandLevelControlClusterStepWithOnOff(*stepMode*, *stepSize*, *transitionTime*)

Command description for Step With On/Off.

Cluster: Level Control, Attributes and commands for controlling devices that can be set to a level between fully 'On' and fully 'Off.' Command: Step With On/Off

Parameters

<i>stepMode</i>	uint8_t
<i>stepSize</i>	uint8_t
<i>transitionTime</i>	uint16_t

Definition at line 502 of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.45 #define emberAfRf4ceZrcHaFillCommandLevelControlClusterStopWithOnOff()

Command description for Stop With On/Off.

Cluster: Level Control, Attributes and commands for controlling devices that can be set to a level between fully 'On' and fully 'Off.' Command: Stop With On/Off

Definition at line 517 of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.46 #define emberAfRf4ceZrcHaFillCommandDoorLockClusterLockDoor(*pinRfidCode*)

Locks the door.

Cluster: Door Lock, Provides an interface into a generic way to secure a door. Command: LockDoor

Parameters

<i>pinRfidCode</i>	uint8_t*
--------------------	----------

Definition at line 531 of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.47 #define emberAfRf4ceZrcHaFillCommandDoorLockClusterUnlockDoor(*pinRfidCode*)

Unlocks the door.

Cluster: Door Lock, Provides an interface into a generic way to secure a door. Command: UnlockDoor

Parameters

<i>pinRfidCode</i>	uint8_t*
--------------------	----------

Definition at line 543 of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.48 #define emberAfRf4ceZrcHaFillCommandDoorLockClusterToggle(*pinRfidCode*)

Toggles the door lock from its current state to the opposite state locked or unlocked.

Cluster: Door Lock, Provides an interface into a generic way to secure a door. Command: Toggle

Parameters

<i>pinRfidCode</i>	uint8_t*
--------------------	----------

Definition at line 555 of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.49 #define emberAfRf4ceZrcHaFillCommandDoorLockClusterUnlockWithTimeout(timeoutInSeconds, pinRfidCode)

Unlock the door with a timeout. When the timeout expires, the door will automatically re-lock.

Cluster: Door Lock, Provides an interface into a generic way to secure a door. Command: UnlockWithTimeout

Parameters

<i>timeoutIn-Seconds</i>	uint16_t
<i>pinRfidCode</i>	uint8_t*

Definition at line [568](#) of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.50 #define emberAfRf4ceZrcHaFillCommandWindowCoveringClusterUpOpen()

Moves window covering to InstalledOpenLimit - Lift and InstalledOpenLimit - Tilt.

Cluster: Window Covering, Provides an interface for controlling and adjusting automatic window coverings. Command: WindowCoveringUpOpen

Definition at line [584](#) of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.51 #define emberAfRf4ceZrcHaFillCommandWindowCoveringClusterDownClose()

Moves window covering to InstalledClosedLimit - Lift and InstalledCloseLimit - Tilt.

Cluster: Window Covering, Provides an interface for controlling and adjusting automatic window coverings. Command: WindowCoveringDownClose

Definition at line [594](#) of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.52 #define emberAfRf4ceZrcHaFillCommandWindowCoveringClusterStop()

Stop any adjusting of window covering.

Cluster: Window Covering, Provides an interface for controlling and adjusting automatic window coverings. Command: WindowCoveringStop

Definition at line [604](#) of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.53 #define emberAfRf4ceZrcHaFillCommandWindowCoveringClusterGoToLiftValue(liftValue)

Goto lift value specified.

Cluster: Window Covering, Provides an interface for controlling and adjusting automatic window coverings. Command: WindowCoveringGoToLiftValue

Parameters

<i>liftValue</i>	uint16_t
------------------	----------

Definition at line [615](#) of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.54 #define emberAfRf4ceZrcHaFillCommandWindowCoveringClusterGoToLiftPercentage(percentageLiftValue)

Goto lift percentage specified.

Cluster: Window Covering, Provides an interface for controlling and adjusting automatic window coverings. Command: WindowCoveringGoToLiftPercentage

Parameters

<i>percentageLift-Value</i>	uint8_t
-----------------------------	---------

Definition at line [627](#) of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.55 #define emberAfRf4ceZrcHaFillCommandWindowCoveringClusterGoToTiltValue(tiltValue)

Goto tilt value specified.

Cluster: Window Covering, Provides an interface for controlling and adjusting automatic window coverings. Command: WindowCoveringGoToTiltValue

Parameters

<i>tiltValue</i>	uint16_t
------------------	----------

Definition at line [639](#) of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.56 #define emberAfRf4ceZrcHaFillCommandWindowCoveringClusterGoToTiltPercentage(percentageTiltValue)

Goto tilt percentage specified.

Cluster: Window Covering, Provides an interface for controlling and adjusting automatic window coverings. Command: WindowCoveringGoToTiltPercentage

Parameters

<i>percentageTilt-Value</i>	uint8_t
-----------------------------	---------

Definition at line [651](#) of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.57 #define emberAfRf4ceZrcHaFillCommandThermostatClusterSetpointRaiseLower(mode, amount)

Command description for SetpointRaiseLower.

Cluster: Thermostat, An interface for configuring and controlling the functionality of a thermostat. Command: SetpointRaiseLower

Parameters

<i>mode</i>	uint8_t
<i>amount</i>	int8_t

Definition at line 667 of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.58 #define emberAfRf4ceZrcHaFillCommandColorControlClusterMoveToHue(*hue*, *direction*, *transitionTime*)

Move to specified hue.

Cluster: Color Control, Attributes and commands for controlling the color properties of a color-capable light. Command: MoveToHue

Parameters

<i>hue</i>	uint8_t
<i>direction</i>	uint8_t
<i>transitionTime</i>	uint16_t

Definition at line 686 of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.59 #define emberAfRf4ceZrcHaFillCommandColorControlClusterMoveHue(*moveMode*, *rate*)

Move hue up or down at specified rate.

Cluster: Color Control, Attributes and commands for controlling the color properties of a color-capable light. Command: MoveHue

Parameters

<i>moveMode</i>	uint8_t
<i>rate</i>	uint8_t

Definition at line 703 of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.60 #define emberAfRf4ceZrcHaFillCommandColorControlClusterStepHue(*stepMode*, *stepSize*, *transitionTime*)

Step hue up or down by specified size at specified rate.

Cluster: Color Control, Attributes and commands for controlling the color properties of a color-capable light. Command: StepHue

Parameters

<i>stepMode</i>	uint8_t
<i>stepSize</i>	uint8_t
<i>transitionTime</i>	uint8_t

Definition at line 719 of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.61 #define emberAfRf4ceZrcHaFillCommandColorControlClusterMoveToSaturation(*saturation*, *transitionTime*)

Move to specified saturation.

Cluster: Color Control, Attributes and commands for controlling the color properties of a color-capable light. Command: MoveToSaturation

Parameters

<i>saturation</i>	uint8_t
<i>transitionTime</i>	uint16_t

Definition at line [736](#) of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.62 #define emberAfRf4ceZrcHaFillCommandColorControlClusterMoveSaturation(*moveMode*, *rate*)

Move saturation up or down at specified rate.

Cluster: Color Control, Attributes and commands for controlling the color properties of a color-capable light. Command: MoveSaturation

Parameters

<i>moveMode</i>	uint8_t
<i>rate</i>	uint8_t

Definition at line [751](#) of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.63 #define emberAfRf4ceZrcHaFillCommandColorControlClusterStepSaturation(*stepMode*, *stepSize*, *transitionTime*)

Step saturation up or down by specified size at specified rate.

Cluster: Color Control, Attributes and commands for controlling the color properties of a color-capable light. Command: StepSaturation

Parameters

<i>stepMode</i>	uint8_t
<i>stepSize</i>	uint8_t
<i>transitionTime</i>	uint8_t

Definition at line [767](#) of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.64 #define emberAfRf4ceZrcHaFillCommandColorControlClusterMoveToHueAndSaturation(*hue*, *saturation*, *transitionTime*)

Move to hue and saturation.

Cluster: Color Control, Attributes and commands for controlling the color properties of a color-capable light. Command: MoveToHueAndSaturation

Parameters

<i>hue</i>	uint8_t
<i>saturation</i>	uint8_t
<i>transitionTime</i>	uint16_t

Definition at line 785 of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.65 #define emberAfRf4ceZrcHaFillCommandColorControlClusterMoveToColor(colorX, colorY, transitionTime)

Move to specified color.

Cluster: Color Control, Attributes and commands for controlling the color properties of a color-capable light. Command: MoveToColor

Parameters

<i>colorX</i>	uint16_t
<i>colorY</i>	uint16_t
<i>transitionTime</i>	uint16_t

Definition at line 803 of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.66 #define emberAfRf4ceZrcHaFillCommandColorControlClusterMoveColor(rateX, rateY)

Moves the color.

Cluster: Color Control, Attributes and commands for controlling the color properties of a color-capable light. Command: MoveColor

Parameters

<i>rateX</i>	int16_t
<i>rateY</i>	int16_t

Definition at line 820 of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.67 #define emberAfRf4ceZrcHaFillCommandColorControlClusterStepColor(stepX, stepY, transitionTime)

Steps the lighting to a specific color.

Cluster: Color Control, Attributes and commands for controlling the color properties of a color-capable light. Command: StepColor

Parameters

<i>stepX</i>	int16_t
<i>stepY</i>	int16_t
<i>transitionTime</i>	uint16_t

Definition at line 836 of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.68 #define emberAfRf4ceZrcHaFillCommandColorControlClusterMoveToColorTemperature(colorTemperature, transitionTime)

Moves the lighting to a specific color temperature.

Cluster: Color Control, Attributes and commands for controlling the color properties of a color-capable light. Command: MoveToColorTemperature

Parameters

<i>colorTemperature</i>	uint16_t
<i>transitionTime</i>	uint16_t

Definition at line 853 of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.69 #define emberAfRf4ceZrcHaFillCommandIASACEClusterArm(armMode)

Command description for Arm.

Cluster: IAS ACE, Attributes and commands for IAS Ancillary Control Equipment. Command: Arm

Parameters

<i>armMode</i>	uint8_t
----------------	---------

Definition at line 870 of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.70 #define emberAfRf4ceZrcHaFillCommandIASACEClusterBypass(numberOfZones, zoneIds, zoneIdsLen)

Command description for Bypass.

Cluster: IAS ACE, Attributes and commands for IAS Ancillary Control Equipment. Command: Bypass

Parameters

<i>numberOfZones</i>	uint8_t
<i>zoneIds</i>	uint8_t*
<i>zoneIdsLen</i>	uint16_t

Definition at line 884 of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.71 #define emberAfRf4ceZrcHaFillCommandIASACEClusterEmergency()

Command description for Emergency.

Cluster: IAS ACE, Attributes and commands for IAS Ancillary Control Equipment. Command: Emergency

Definition at line 899 of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.72 #define emberAfRf4ceZrcHaFillCommandIASACEClusterFire()

Command description for Fire.

Cluster: IAS ACE, Attributes and commands for IAS Ancillary Control Equipment. Command: Fire

Definition at line 909 of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.73 #define emberAfRf4ceZrcHaFillCommandIASACEClusterPanic()

Command description for Panic.

Cluster: IAS ACE, Attributes and commands for IAS Ancillary Control Equipment. Command: Panic

Definition at line 919 of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.74 #define emberAfRf4ceZrcHaFillCommandPreviousDestinationGroup()

Previous destination group.

Command: PreviousDestinationGroup

Definition at line 929 of file [rf4ce-zrc20-ha-client.h](#).

6.13.2.75 #define emberAfRf4ceZrcHaFillCommandNextDestinationGroup()

Next destination group.

Command: NextDestinationGroup

Definition at line 938 of file [rf4ce-zrc20-ha-client.h](#).

6.13.3 Enumeration Type Documentation

6.13.3.1 enum EmberAfThermostatSetpointRaiseLowerMode

Enumerator:

```
EMBER_ZCL_THERMOSTAT_SETPOINT_RAISE_LOWER_MODE_HEAT
EMBER_ZCL_THERMOSTAT_SETPOINT_RAISE_LOWER_MODE_COOL
EMBER_ZCL_THERMOSTAT_SETPOINT_RAISE_LOWER_MODE_BOTH
```

Definition at line 44 of file [rf4ce-zrc20-ha-client.h](#).

6.13.4 Function Documentation

6.13.4.1 uint16_t emAfPluginRf4ceZrc20HaFillExternalBuffer (PGM_P *format*, ...)

6.13.4.2 EmberStatus emberAfRf4ceZrcHaSend (uint8_t *pairingIndex*, uint8_t *haInstanceId*)

Send HA command to the server.

Parameters

<i>pairingIndex</i>	The pairing index the send HA command should be sent to.
<i>haInstanceId</i>	The HA instance ID the send HA command should be sent to.

Returns

An `:EmberStatus` value indicating whether the send HA command was successfully sent out or the reason of failure.

6.14 ZigBee Remote Control 2.0 Home Automation Server

Data Structures

- struct DestStruct
- struct HaAttributesInfo

Macros

- #define EMBER_AF_PLUGIN_RF4CE_ZRC20_HA_SERVER_ZCL_BUFFER_SIZE
- #define ZRC_HA_SERVER_NUM_OF_HA_INSTANCES
- #define ZRC_ACTION_ID_HIGH_NIBBLE_MASK
- #define ZRC_ACTION_ID_LOW_NIBBLE_MASK
- #define ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE0_ID
- #define ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE1_ID
- #define ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE2_ID
- #define ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE3_ID
- #define ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE4_ID
- #define ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE5_ID
- #define ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE6_ID
- #define ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE7_ID
- #define ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE8_ID
- #define ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE9_ID
- #define ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE10_ID
- #define ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE11_ID
- #define ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE12_ID
- #define ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE13_ID
- #define ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE14_ID
- #define ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE15_ID
- #define ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE_ID_SIZE
- #define ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE_SIZE
- #define ZRC_HA_SCENE_LOCAL_SCENE_VALID0_ID
- #define ZRC_HA_SCENE_LOCAL_SCENE_VALID1_ID
- #define ZRC_HA_SCENE_LOCAL_SCENE_VALID2_ID
- #define ZRC_HA_SCENE_LOCAL_SCENE_VALID3_ID
- #define ZRC_HA_SCENE_LOCAL_SCENE_VALID4_ID
- #define ZRC_HA_SCENE_LOCAL_SCENE_VALID5_ID
- #define ZRC_HA_SCENE_LOCAL_SCENE_VALID6_ID
- #define ZRC_HA_SCENE_LOCAL_SCENE_VALID7_ID
- #define ZRC_HA_SCENE_LOCAL_SCENE_VALID8_ID
- #define ZRC_HA_SCENE_LOCAL_SCENE_VALID9_ID
- #define ZRC_HA_SCENE_LOCAL_SCENE_VALID10_ID
- #define ZRC_HA_SCENE_LOCAL_SCENE_VALID11_ID
- #define ZRC_HA_SCENE_LOCAL_SCENE_VALID12_ID
- #define ZRC_HA_SCENE_LOCAL_SCENE_VALID13_ID
- #define ZRC_HA_SCENE_LOCAL_SCENE_VALID14_ID
- #define ZRC_HA_SCENE_LOCAL_SCENE_VALID15_ID
- #define ZRC_HA_SCENE_LOCAL_SCENE_VALID_ID_SIZE
- #define ZRC_HA_SCENE_LOCAL_SCENE_VALID_SIZE
- #define ZRC_HA_ON_OFF_ON_OFF_ID

- #define ZRC_HA_ON_OFF_ON_OFF_ID_SIZE
- #define ZRC_HA_ON_OFF_ON_OFF_SIZE
- #define ZRC_HA_LEVEL_CONTROL_CURRENT_LEVEL_ID
- #define ZRC_HA_LEVEL_CONTROL_CURRENT_LEVEL_ID_SIZE
- #define ZRC_HA_LEVEL_CONTROL_CURRENT_LEVEL_SIZE
- #define ZRC_HA_DOOR_LOCK_LOCK_DOOR_RESPONSE_ID
- #define ZRC_HA_DOOR_LOCK_LOCK_DOOR_RESPONSE_ID_SIZE
- #define ZRC_HA_DOOR_LOCK_LOCK_DOOR_RESPONSE_SIZE
- #define ZRC_HA_DOOR_LOCK_UNLOCK_DOOR_RESPONSE_ID
- #define ZRC_HA_DOOR_LOCK_UNLOCK_DOOR_RESPONSE_ID_SIZE
- #define ZRC_HA_DOOR_LOCK_UNLOCK_DOOR_RESPONSE_SIZE
- #define ZRC_HA_DOOR_LOCK_TOGGLE_RESPONSE_ID
- #define ZRC_HA_DOOR_LOCK_TOGGLE_RESPONSE_ID_SIZE
- #define ZRC_HA_DOOR_LOCK_TOGGLE_RESPONSE_SIZE
- #define ZRC_HA_DOOR_LOCK_UNLOCK_WITH_TIMEOUT_RESPONSE_ID
- #define ZRC_HA_DOOR_LOCK_UNLOCK_WITH_TIMEOUT_RESPONSE_ID_SIZE
- #define ZRC_HA_DOOR_LOCK_UNLOCK_WITH_TIMEOUT_RESPONSE_SIZE
- #define ZRC_HA_WINDOW_COVERING_CURRENT_POSITION_LIFT_PERCENTAGE_ID
- #define ZRC_HA_WINDOW_COVERING_CURRENT_POSITION_LIFT_PERCENTAGE_ID_SIZE
- #define ZRC_HA_WINDOW_COVERING_CURRENT_POSITION_LIFT_PERCENTAGE_SIZE
- #define ZRC_HA_WINDOW_COVERING_CURRENT_POSITION_TILT_PERCENTAGE_ID
- #define ZRC_HA_WINDOW_COVERING_CURRENT_POSITION_TILT_PERCENTAGE_ID_SIZE
- #define ZRC_HA_WINDOW_COVERING_CURRENT_POSITION_TILT_PERCENTAGE_SIZE
- #define ZRC_HA_THERMOSTAT_LOCAL_TEMPERATURE_ID
- #define ZRC_HA_THERMOSTAT_LOCAL_TEMPERATURE_ID_SIZE
- #define ZRC_HA_THERMOSTAT_LOCAL_TEMPERATURE_SIZE
- #define ZRC_HA_THERMOSTAT_OCCUPIED_COOLING_SETPOINT_ID
- #define ZRC_HA_THERMOSTAT_OCCUPIED_COOLING_SETPOINT_ID_SIZE
- #define ZRC_HA_THERMOSTAT_OCCUPIED_COOLING_SETPOINT_SIZE
- #define ZRC_HA_THERMOSTAT_OCCUPIED_HEATING_SETPOINT_ID
- #define ZRC_HA_THERMOSTAT_OCCUPIED_HEATING_SETPOINT_ID_SIZE
- #define ZRC_HA_THERMOSTAT_OCCUPIED_HEATING_SETPOINT_SIZE
- #define ZRC_HA_THERMOSTAT_OCCUPIED_HEATING_SETPOINT_SIZE
- #define ZRC_HA_COLOR_CONTROL_CURRENT_HUE_ID
- #define ZRC_HA_COLOR_CONTROL_CURRENT_HUE_ID_SIZE
- #define ZRC_HA_COLOR_CONTROL_CURRENT_HUE_SIZE
- #define ZRC_HA_COLOR_CONTOL_CURRENT_SATURATION_ID
- #define ZRC_HA_COLOR_CONTOL_CURRENT_SATURATION_ID_SIZE
- #define ZRC_HA_COLOR_CONTOL_CURRENT_SATURATION_SIZE
- #define ZRC_HA_COLOR_CONTOL_CURRENT_X_ID
- #define ZRC_HA_COLOR_CONTOL_CURRENT_X_ID_SIZE
- #define ZRC_HA_COLOR_CONTOL_CURRENT_X_SIZE
- #define ZRC_HA_COLOR_CONTOL_CURRENT_Y_ID
- #define ZRC_HA_COLOR_CONTOL_CURRENT_Y_ID_SIZE
- #define ZRC_HA_COLOR_CONTOL_CURRENT_Y_SIZE
- #define ZRC_HA_COLOR_CONTOL_COLOR_MODE_ID
- #define ZRC_HA_COLOR_CONTOL_COLOR_MODE_ID_SIZE

- #define ZRC_HA_COLOR_CONTOL_COLOR_MODE_SIZE
- #define ZRC_HA_IAS_ACE_ARM_RESPONSE_ID
- #define ZRC_HA_IAS_ACE_ARM_RESPONSE_ID_SIZE
- #define ZRC_HA_IAS_ACE_ARM_RESPONSE_SIZE
- #define ZRC_HA_ATTRIBUTE_STATUS_TABLE_SIZE
- #define ZRC_HA_ATTRIBUTE_TABLE_SIZE
- #define HA_ATTRIBUTE_STATUS_LENGTH
- #define HA_ATTRIBUTE_STATUS_CHANGED_FLAG

Functions

- void `emberAfPluginRf4ceZrc20HaServerClearAllHaAttributes` (void)
- EmberAfRf4ceGdpAttributeStatus `emberAfPluginRf4ceZrc20HaServerGetHaAttribute` (uint8_t pairingIndex, uint8_t haInstanceId, uint8_t haAttributeId, `EmberAfRf4ceZrcHomeAutomationAttribute` *haAttribute)
- EmberAfRf4ceGdpAttributeStatus `emberAfPluginRf4ceZrc20HaServerSetHaAttribute` (uint8_t pairingIndex, uint8_t haInstanceId, uint8_t haAttributeId, `EmberAfRf4ceZrcHomeAutomationAttribute` *haAttribute)
- EmberStatus `emberAfPluginRf4ceZrc20HaMappingToLogicalDeviceAdd` (uint8_t pairingIndex, uint8_t haInstanceId, uint8_t destIndex)
- EmberStatus `emberAfPluginRf4ceZrc20HaMappingToLogicalDeviceRemove` (uint8_t destIndex)
- EmberStatus `emberAfPluginRf4ceZrc20HaMappingToLogicalDeviceGet` (uint8_t pairingIndex, uint8_t haInstanceId, uint8_t *destIndex)
- EmberStatus `emberAfPluginRf4ceZrc20HaLogicalDeviceDestinationAdd` (`DestStruct` *dest, uint8_t *index)
- EmberStatus `emberAfPluginRf4ceZrc20HaLogicalDeviceDestinationRemove` (`DestStruct` *dest, uint8_t *index)
- uint8_t `emberAfPluginRf4ceZrc20HaLogicalDeviceDestinationTableSize` (void)
- EmberStatus `emberAfPluginRf4ceZrc20HaLogicalDeviceDestinationGet` (uint8_t pairingIndex, uint8_t haInstanceId, `DestStruct` *dest)
- EmberStatus `emberAfPluginRf4ceZrc20HaLogicalDeviceIndexLookUp` (`DestStruct` *dest, uint8_t *index)
- void `emberAfPluginRf4ceZrc20HaLogicalDeviceAndInstanceToLogicalDeviceMappingClear` (void)
- EmberStatus `emberAfPluginRf4ceZrc20HaLogicalDeviceAndInstanceToLogicalDeviceMappingAdd` (uint8_t pairingIndex, uint8_t haInstanceId, `DestStruct` *dest)
- EmberStatus `emberAfPluginRf4ceZrc20HaLogicalDeviceAndInstanceToLogicalDeviceMappingRemove` (`DestStruct` *dest)
- EmberStatus `emAfRf4ceZrc20ParseHaActionAndForwardToZclNetwork` (const `EmberAfRf4ceZrcActionRecord` *record)
- void `emAfRf4ceZrc20ClearLogicalDevicesTable` (void)
- void `emAfRf4ceZrc20ClearInstanceToLogicalDeviceTable` (void)
- EmberStatus `emAfRf4ceZrc20AddLogicalDeviceDestination` (`DestStruct` *dest, uint8_t *index)
- EmberStatus `emAfRf4ceZrc20RemoveLogicalDeviceDestination` (uint8_t destIndex)
- uint8_t `GetLogicalDeviceDestination` (uint8_t i, `DestStruct` *dest)
- void `DestLookup` (uint8_t pairingIndex, uint8_t haInstanceId, `DestStruct` *dest)

Variables

- EmberOutgoingMessageType `DestStruct::type`
- uint16_t `DestStruct::indexOrDestination`
- uint8_t `DestStruct::sourceEndpoint`
- uint8_t `DestStruct::destinationEndpoint`
- uint8_t `HaAttributesInfo::id`
- uint8_t `HaAttributesInfo::length`

6.14.1 Detailed Description

The ZigBee Remote Control 2.0 (ZRC2.0) Home Automation (HA) Server plugin implements the optional HA interoperability feature of the ZRC2.0 profile for HA action recipients. HA interoperability allows RF4CE devices to control HA devices operating on a ZigBee PRO network within the same premises.

HA action originators are most typically controllers but may also be targets. HA action recipients must be targets. HA originators transmit HA commands to an HA recipient over the RF4CE network as ZRC2.0 actions. The recipient then translates these actions to ZigBee Cluster Library (ZCL) messages and retransmits them to the appropriate HA device on the PRO network. The originator directs all actions to abstract HA instances. Each HA action recipient is responsible for mapping HA instances to actual physical devices.

Users will want to take note of the callback `emberAfPluginZrc20HaServerHaActionSentCallback` (see [Ember RF4CE Application Framework Callbacks](#) for more information regarding callbacks). This callback conveys to the application that the HA server has sent an HA action to a ZCL network.

Furthermore, customers can use this plugin to manage an application's store of HA Attributes and Logical Device information. There are functions to interface with the database of these items, as well as definitions of constants for an application to use.

This plugin utilizes the action functionality provided by the [ZigBee Remote Control 2.0 Profile](#) plugin. Support for the optional HA action originator feature is provided by the [ZigBee Remote Control 2.0 Home Automation Client](#) plugin.

6.14.2 Macro Definition Documentation

6.14.2.1 #define EMBER_AF_PLUGIN_RF4CE_ZRC20_HA_SERVER_ZCL_BUFFER_SIZE

Definition at line 43 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.2 #define ZRC_HA_SERVER_NUM_OF_HA_INSTANCES

Definition at line 45 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.3 #define ZRC_ACTION_ID_HIGH_NIBBLE_MASK

Definition at line 47 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.4 #define ZRC_ACTION_ID_LOW_NIBBLE_MASK

Definition at line 48 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.5 #define ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE0_ID

Definition at line 53 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.6 #define ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE1_ID

Definition at line 54 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.7 #define ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE2_ID

Definition at line 55 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.8 #define ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE3_ID

Definition at line 56 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.9 #define ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE4_ID

Definition at line 57 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.10 #define ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE5_ID

Definition at line 58 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.11 #define ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE6_ID

Definition at line 59 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.12 #define ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE7_ID

Definition at line 60 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.13 #define ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE8_ID

Definition at line 61 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.14 #define ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE9_ID

Definition at line 62 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.15 #define ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE10_ID

Definition at line 63 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.16 #define ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE11_ID

Definition at line 64 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.17 #define ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE12_ID

Definition at line 65 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.18 #define ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE13_ID

Definition at line 66 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.19 #define ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE14_ID

Definition at line 67 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.20 #define ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE15_ID

Definition at line 68 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.21 #define ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE_ID_SIZE

Definition at line 69 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.22 #define ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE_SIZE

Definition at line 70 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.23 #define ZRC_HA_SCENE_LOCAL_SCENE_VALID0_ID

Definition at line 71 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.24 #define ZRC_HA_SCENE_LOCAL_SCENE_VALID1_ID

Definition at line 72 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.25 #define ZRC_HA_SCENE_LOCAL_SCENE_VALID2_ID

Definition at line 73 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.26 #define ZRC_HA_SCENE_LOCAL_SCENE_VALID3_ID

Definition at line 74 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.27 #define ZRC_HA_SCENE_LOCAL_SCENE_VALID4_ID

Definition at line 75 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.28 #define ZRC_HA_SCENE_LOCAL_SCENE_VALID5_ID

Definition at line 76 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.29 #define ZRC_HA_SCENE_LOCAL_SCENE_VALID6_ID

Definition at line 77 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.30 #define ZRC_HA_SCENE_LOCAL_SCENE_VALID7_ID

Definition at line 78 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.31 #define ZRC_HA_SCENE_LOCAL_SCENE_VALID8_ID

Definition at line 79 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.32 #define ZRC_HA_SCENE_LOCAL_SCENE_VALID9_ID

Definition at line 80 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.33 #define ZRC_HA_SCENE_LOCAL_SCENE_VALID10_ID

Definition at line 81 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.34 #define ZRC_HA_SCENE_LOCAL_SCENE_VALID11_ID

Definition at line 82 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.35 #define ZRC_HA_SCENE_LOCAL_SCENE_VALID12_ID

Definition at line 83 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.36 #define ZRC_HA_SCENE_LOCAL_SCENE_VALID13_ID

Definition at line 84 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.37 #define ZRC_HA_SCENE_LOCAL_SCENE_VALID14_ID

Definition at line 85 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.38 #define ZRC_HA_SCENE_LOCAL_SCENE_VALID15_ID

Definition at line 86 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.39 #define ZRC_HA_SCENE_LOCAL_SCENE_VALID_ID_SIZE

Definition at line 87 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.40 #define ZRC_HA_SCENE_LOCAL_SCENE_VALID_SIZE

Definition at line 88 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.41 #define ZRC_HA_ON_OFF_ON_OFF_ID

Definition at line 89 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.42 #define ZRC_HA_ON_OFF_ON_OFF_ID_SIZE

Definition at line 90 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.43 #define ZRC_HA_ON_OFF_ON_OFF_SIZE

Definition at line 91 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.44 #define ZRC_HA_LEVEL_CONTROL_CURRENT_LEVEL_ID

Definition at line 92 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.45 #define ZRC_HA_LEVEL_CONTROL_CURRENT_LEVEL_ID_SIZE

Definition at line 93 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.46 #define ZRC_HA_LEVEL_CONTROL_CURRENT_LEVEL_SIZE

Definition at line 94 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.47 #define ZRC_HA_DOOR_LOCK_LOCK_DOOR_RESPONSE_ID

Definition at line 95 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.48 #define ZRC_HA_DOOR_LOCK_LOCK_DOOR_RESPONSE_ID_SIZE

Definition at line 96 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.49 #define ZRC_HA_DOOR_LOCK_LOCK_DOOR_RESPONSE_SIZE

Definition at line 97 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.50 #define ZRC_HA_DOOR_LOCK_UNLOCK_DOOR_RESPONSE_ID

Definition at line 98 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.51 #define ZRC_HA_DOOR_LOCK_UNLOCK_DOOR_RESPONSE_ID_SIZE

Definition at line 99 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.52 #define ZRC_HA_DOOR_LOCK_UNLOCK_DOOR_RESPONSE_SIZE

Definition at line 100 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.53 #define ZRC_HA_DOOR_LOCK_TOGGLE_RESPONSE_ID

Definition at line 101 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.54 #define ZRC_HA_DOOR_LOCK_TOGGLE_RESPONSE_ID_SIZE

Definition at line 102 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.55 #define ZRC_HA_DOOR_LOCK_TOGGLE_RESPONSE_SIZE

Definition at line 103 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.56 #define ZRC_HA_DOOR_LOCK_UNLOCK_WITH_TIMEOUT_RESPONSE_ID

Definition at line 104 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.57 #define ZRC_HA_DOOR_LOCK_UNLOCK_WITH_TIMEOUT_RESPONSE_ID_SIZE

Definition at line 105 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.58 #define ZRC_HA_DOOR_LOCK_UNLOCK_WITH_TIMEOUT_RESPONSE_SIZE

Definition at line 106 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.59 #define ZRC_HA_WINDOW_COVERING_CURRENT_POSITION_LIFT_PERCENTAGE_ID

Definition at line 107 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.60 #define ZRC_HA_WINDOW_COVERING_CURRENT_POSITION_LIFT_PERCENTAGE_ID_SIZE

Definition at line 108 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.61 #define ZRC_HA_WINDOW_COVERING_CURRENT_POSITION_LIFT_PERCENTAGE_SIZE

Definition at line 109 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.62 #define ZRC_HA_WINDOW_COVERING_CURRENT_POSITION_TILT_PERCENTAGE_ID

Definition at line 110 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.63 #define ZRC_HA_WINDOW_COVERING_CURRENT_POSITION_TILT_PERCENTAGE_ID_SIZE

Definition at line 111 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.64 #define ZRC_HA_WINDOW_COVERING_CURRENT_POSITION_TILT_PERCENTAGE_SIZE

Definition at line 112 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.65 #define ZRC_HA_THERMOSTAT_LOCAL_TEMPERATURE_ID

Definition at line 113 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.66 #define ZRC_HA_THERMOSTAT_LOCAL_TEMPERATURE_ID_SIZE

Definition at line 114 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.67 #define ZRC_HA_THERMOSTAT_LOCAL_TEMPERATURE_SIZE

Definition at line 115 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.68 #define ZRC_HA_THERMOSTAT_OCCUPIED_COOLING_SETPOINT_ID

Definition at line 116 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.69 #define ZRC_HA_THERMOSTAT_OCCUPIED_COOLING_SETPOINT_ID_SIZE

Definition at line 117 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.70 #define ZRC_HA_THERMOSTAT_OCCUPIED_COOLING_SETPOINT_SIZE

Definition at line 118 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.71 #define ZRC_HA_THERMOSTAT_OCCUPIED_HEATING_SETPOINT_ID

Definition at line 119 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.72 #define ZRC_HA_THERMOSTAT_OCCUPIED_HEATING_SETPOINT_ID_SIZE

Definition at line 120 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.73 #define ZRC_HA_THERMOSTAT_OCCUPIED_HEATING_SETPOINT_SIZE

Definition at line 121 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.74 #define ZRC_HA_COLOR_CONTROL_CURRENT_HUE_ID

Definition at line 122 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.75 #define ZRC_HA_COLOR_CONTROL_CURRENT_HUE_ID_SIZE

Definition at line 123 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.76 #define ZRC_HA_COLOR_CONTROL_CURRENT_HUE_SIZE

Definition at line 124 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.77 #define ZRC_HA_COLOR_CONTOL_CURRENT_SATURATION_ID

Definition at line 125 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.78 #define ZRC_HA_COLOR_CONTOL_CURRENT_SATURATION_ID_SIZE

Definition at line 126 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.79 #define ZRC_HA_COLOR_CONTOL_CURRENT_SATURATION_SIZE

Definition at line 127 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.80 #define ZRC_HA_COLOR_CONTOL_CURRENT_X_ID

Definition at line 128 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.81 #define ZRC_HA_COLOR_CONTOL_CURRENT_X_ID_SIZE

Definition at line 129 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.82 #define ZRC_HA_COLOR_CONTOL_CURRENT_X_SIZE

Definition at line 130 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.83 #define ZRC_HA_COLOR_CONTOL_CURRENT_Y_ID

Definition at line 131 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.84 #define ZRC_HA_COLOR_CONTOL_CURRENT_Y_ID_SIZE

Definition at line 132 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.85 #define ZRC_HA_COLOR_CONTOL_CURRENT_Y_SIZE

Definition at line 133 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.86 #define ZRC_HA_COLOR_CONTOL_COLOR_MODE_ID

Definition at line 134 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.87 #define ZRC_HA_COLOR_CONTOL_COLOR_MODE_ID_SIZE

Definition at line 135 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.88 #define ZRC_HA_COLOR_CONTOL_COLOR_MODE_SIZE

Definition at line 136 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.89 #define ZRC_HA_IAS_ACE_ARM_RESPONSE_ID

Definition at line 137 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.90 #define ZRC_HA_IAS_ACE_ARM_RESPONSE_ID_SIZE

Definition at line 138 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.91 #define ZRC_HA_IAS_ACE_ARM_RESPONSE_SIZE

Definition at line 139 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.92 #define ZRC_HA_ATTRIBUTE_STATUS_TABLE_SIZE

Definition at line 141 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.93 #define ZRC_HA_ATTRIBUTE_TABLE_SIZE

Definition at line 162 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.94 #define HA_ATTRIBUTE_STATUS_LENGTH

Definition at line 183 of file [rf4ce-zrc20-ha-server.h](#).

6.14.2.95 #define HA_ATTRIBUTE_STATUS_CHANGED_FLAG

Definition at line 188 of file [rf4ce-zrc20-ha-server.h](#).

6.14.3 Function Documentation

6.14.3.1 void emberAfPluginRf4ceZrc20HaServerClearAllHaAttributes (void)

Clear all HA attributes on the server.

6.14.3.2 EmberAfRf4ceGdpAttributeStatus emberAfPluginRf4ceZrc20HaServerGetHaAttribute (uint8_t pairingIndex, uint8_t haInstanceId, uint8_t haAttributeId, EmberAfRf4ceZrcHomeAutomationAttribute * haAttribute)

Get selected HA attribute on the server.

Parameters

<i>pairingIndex</i>	The pairing index the get HA attribute command should be sent to.
<i>haInstanceId</i>	The instance ID of the HA attribute to get.
<i>haAttributeId</i>	The attribute ID of the HA attribute to get.
<i>haAttribute</i>	The HA attribute structure describing the HA attribute.

Returns

An [EmberAfRf4ceGdpAttributeStatus](#) value indicating whether the get HA attribute command was successfully sent out or the reason of failure.

6.14.3.3 EmberAfRf4ceGdpAttributeStatus emberAfPluginRf4ceZrc20HaServerSetHaAttribute (uint8_t pairingIndex, uint8_t haInstanceId, uint8_t haAttributeId, EmberAfRf4ceZrcHomeAutomationAttribute * haAttribute)

Set selected HA attribute on the server.

Parameters

<i>pairingIndex</i>	The pairing index the set HA attribute command should be sent to.
<i>haInstanceId</i>	The instance ID of the HA attribute to set.
<i>haAttributeId</i>	The attribute ID of the HA attribute to set.
<i>haAttribute</i>	The HA attribute structure describing the HA attribute.

Returns

An [EmberAfRf4ceGdpAttributeStatus](#) value indicating whether the set HA attribute command was successfully sent out or the reason of failure.

6.14.3.4 EmberStatus emberAfPluginRf4ceZrc20HaMappingToLogicalDeviceAdd (*uint8_t pairingIndex, uint8_t haInstanceId, uint8_t destIndex*)

Add mapping to HA logical device on the server.

Parameters

<i>pairingIndex</i>	The pairing index to which the HA logical device is mapped.
<i>haInstanceId</i>	The instance ID to which the HA logical device is mapped.
<i>destIndex</i>	The index of the HA logical device in the logical devices table.

Returns

An ::EmberStatus value indicating whether adding mapping to the HA logical device was successfully sent out or the reason of failure.

6.14.3.5 EmberStatus emberAfPluginRf4ceZrc20HaMappingToLogicalDeviceRemove (*uint8_t destIndex*)

Remove mapping to HA logical device on the server.

Parameters

<i>destIndex</i>	The index of the HA logical device in the logical devices table.
------------------	--

Returns

An ::EmberStatus value indicating whether removing mapping to the HA logical device was successfully sent out or the reason of failure.

6.14.3.6 EmberStatus emberAfPluginRf4ceZrc20HaMappingToLogicalDeviceGet (*uint8_t pairingIndex, uint8_t haInstanceId, uint8_t * destIndex*)

Get mapping to HA logical device on the server.

Parameters

<i>pairingIndex</i>	The pairing index to which the HA logical device is mapped.
<i>haInstanceId</i>	The instance ID to which the HA logical device is mapped.
<i>destIndex</i>	The index of the HA logical device in the logical devices table.

Returns

An ::EmberStatus value indicating whether getting mapping to the HA logical device was successfully sent out or the reason of failure.

6.14.3.7 EmberStatus emberAfPluginRf4ceZrc20HaLogicalDeviceDestinationAdd (DestStruct * dest, uint8_t * index)

Add HA logical device to the logical devices table on the server.

Parameters

<i>dest</i>	The destination structure describing the logical device to add.
<i>index</i>	The index of the logical device table to which the HA logical device was added.

Returns

An ::EmberStatus value indicating whether adding logical device was successfully sent out or the reason of failure.

6.14.3.8 EmberStatus emberAfPluginRf4ceZrc20HaLogicalDeviceDestinationRemove (DestStruct * dest, uint8_t * index)

Remove HA logical device from the logical devices table on the server.

Parameters

<i>dest</i>	The destination structure describing the logical device to remove.
<i>index</i>	The index of the logical device table from which the HA logical device was removed.

Returns

An ::EmberStatus value indicating whether removing logical device was successfully sent out or the reason of failure.

6.14.3.9 uint8_t emberAfPluginRf4ceZrc20HaLogicalDeviceDestinationTableSize (void)

Get the size of the HA logical devices table on the server.

Returns

Size of the HA logical devices table.

6.14.3.10 EmberStatus emberAfPluginRf4ceZrc20HaLogicalDeviceDestinationGet (uint8_t pairingIndex, uint8_t haInstanceId, DestStruct * dest)

Get the destination of the HA logical device on the server.

Parameters

<i>pairingIndex</i>	The pairing index to which the HA logical device is mapped.
<i>haInstanceId</i>	The instance ID to which the HA logical device is mapped.
<i>dest</i>	The destination structure describing the HA logical device to get.

Returns

An ::EmberStatus value indicating whether getting HA logical device destination was successfully sent out or the reason of failure.

6.14.3.11 EmberStatus emberAfPluginRf4ceZrc20HaLogicalDeviceIndexLookUp (DestStruct * dest, uint8_t * index)

Look up the index of the HA logical device on the server.

Parameters

<i>dest</i>	The destination structure describing the HA logical device to look up.
<i>index</i>	The index of the HA logical device in the logical devices table.

Returns

An ::EmberStatus value indicating whether looking up HA logical device destination was successfully sent out or the reason of failure.

6.14.3.12 void emberAfPluginRf4ceZrc20HaLogicalDeviceAndInstanceToLogicalDeviceMappingClear (void)

Clear all HA logical devices and mappings to logical devices.

6.14.3.13 EmberStatus emberAfPluginRf4ceZrc20HaLogicalDeviceAndInstanceToLogicalDeviceMappingAdd (uint8_t pairingIndex, uint8_t haInstanceId, DestStruct * dest)

Add HA logical device and map it to pairing index and HA instance ID.

Parameters

<i>pairingIndex</i>	The pairing index to which the HA logical device will be mapped.
<i>haInstanceId</i>	The instance ID to which the HA logical device will be mapped.
<i>dest</i>	The destination structure describing the HA logical device to map.

Returns

An ::EmberStatus value indicating whether mapping the HA logical device was successfully sent out or the reason of failure.

6.14.3.14 EmberStatus emberAfPluginRf4ceZrc20HaLogicalDeviceAndInstanceToLogicalDeviceMappingRemove (DestStruct * dest)

Remove HA logical device and its mappings.

Parameters

<i>dest</i>	The destination structure describing the HA logical device to remove.
-------------	---

Returns

An ::EmberStatus value indicating whether removing the HA logical device was successfully sent out or the reason of failure.

- 6.14.3.15 EmberStatus emAfRf4ceZrc20ParseHaActionAndForwardToZclNetwork (const EmberAfRf4ceZrcActionRecord * *record*)
- 6.14.3.16 void emAfRf4ceZrc20ClearLogicalDevicesTable (void)
- 6.14.3.17 void emAfRf4ceZrc20ClearInstanceToLogicalDeviceTable (void)
- 6.14.3.18 EmberStatus emAfRf4ceZrc20AddLogicalDeviceDestination (DestStruct * *dest*, uint8_t * *index*)
- 6.14.3.19 EmberStatus emAfRf4ceZrc20RemoveLogicalDeviceDestination (uint8_t *destIndex*)
- 6.14.3.20 uint8_t GetLogicalDeviceDestination (uint8_t *i*, DestStruct * *dest*)
- 6.14.3.21 void DestLookup (uint8_t *pairingIndex*, uint8_t *halInstanceId*, DestStruct * *dest*)

6.14.4 Variable Documentation

6.14.4.1 EmberOutgoingMessageType DestStruct::type

Definition at line 193 of file [rf4ce-zrc20-ha-server.h](#).

6.14.4.2 uint16_t DestStruct::indexOrDestination

Definition at line 194 of file [rf4ce-zrc20-ha-server.h](#).

6.14.4.3 uint8_t DestStruct::sourceEndpoint

Definition at line 195 of file [rf4ce-zrc20-ha-server.h](#).

6.14.4.4 uint8_t DestStruct::destinationEndpoint

Definition at line 196 of file [rf4ce-zrc20-ha-server.h](#).

6.14.4.5 uint8_t HaAttributesInfo::id

Definition at line 201 of file [rf4ce-zrc20-ha-server.h](#).

6.14.4.6 uint8_t HaAttributesInfo::length

Definition at line 202 of file [rf4ce-zrc20-ha-server.h](#).

6.15 Ember RF4CE Application Framework Callbacks

Modules

- [Framework Callbacks](#)
- [bulb-pwm-driver Callbacks](#)
- [Button Callbacks](#)
- [Debug Basic Library Callbacks](#)
- [EZSP Common Callbacks](#)
- [HAL Library Callbacks](#)
- [Idle/Sleep Callbacks](#)
- [Low Voltage Shutdown Callbacks](#)
- [Main Callbacks](#)
- [Microphone Codec MSADPCM Callbacks](#)
- [Microphone IMAADPCM Callbacks](#)
- [Generic Device Profile Callbacks](#)
- [RF4CE Multiple System Operators Profile Callbacks](#)
- [RF4CE Profile Support Callbacks](#)
- [RF4CE Stack Library Callbacks](#)
- [RF4CE Target Communication. Callbacks](#)
- [ZigBee Remote Control 1.1 Profile Callbacks](#)
- [ZigBee Remote Control 2.0 Profile Callbacks](#)
- [Stack Minimal Library Callbacks](#)
- [STM32F103RET Library Callbacks](#)
- [sim-eeprom API Callbacks](#)

6.15.1 Detailed Description

The Ember ZigBee RF4CE Application Framework consists of a single framework callback (see [Framework Callbacks](#)) as well as a number of plugin-contributed callbacks. The plugin-contributed callbacks are only available to an application when the plugin is enabled.

6.16 Framework Callbacks

Functions

- int `main` (MAIN_FUNCTION_PARAMETERS)

6.16.1 Detailed Description

These callbacks are contributed by the framework.

6.16.2 Function Documentation

6.16.2.1 int main (MAIN_FUNCTION_PARAMETERS)

Main Application Entry Point.

This is the main application entry point. All applications must implement this function.

6.17 bulb-pwm-driver Callbacks

Functions

- `uint16_t halBulbPwmDriverFrequencyCallback (void)`
- `void halBulbPwmDriverInitCompleteCallback (void)`
- `void halBulbPwmDriverBlinkOnCallback (void)`
- `void halBulbPwmDriverBlinkOffCallback (void)`
- `void halBulbPwmDriverBlinkStartCallback (void)`
- `void halBulbPwmDriverBlinkStopCallback (void)`

6.17.1 Detailed Description

These callbacks are contributed by the bulb-pwm-driver plugin.

6.17.2 Function Documentation

6.17.2.1 `uint16_t halBulbPwmDriverFrequencyCallback (void)`

A callback used to configure the frequency of the PWM driver. This is called by the bulb-pwm driver upon initialization to determine the frequency at which the PWM driver should be driven. It should return either the frequency, in Hz, or `USE_DEFAULT_FREQUENCY` to indicate that the plugin should use the default value. The default value is 1000 Hz, but can be overridden by a macro in the board header if a user wishes.

Application Usage:

Should be implemented by an application layer configuration plugin.

Definition at line 43 of file [callback.doc](#).

6.17.2.2 `void halBulbPwmDriverInitCompleteCallback (void)`

Function to indicate that the PWM driver has been initialized and the bulb should drive the initial LED PWM values at this time.

Application Usage:

Should be implemented by an application layer configuration plugin.

Definition at line 55 of file [callback.doc](#).

6.17.2.3 `void halBulbPwmDriverBlinkOnCallback (void)`

This callback is generated during blinking behavior when it is time to turn the bulb on. While the plugin will determine when to blink the bulb on or off, it is up to this callback to determine how to turn the bulb on.

Application Usage:

Should be implemented by an application layer configuration plugin.

Definition at line 68 of file [callback.doc](#).

6.17.2.4 void halBulbPwmDriverBlinkOffCallback (void)

This callback is generated during blinking behavior when it is time to turn the bulb off. While the plugin will determine when to blink the bulb on or off, it is up to this callback to determine how to turn the bulb off.

Application Usage:

Should be implemented by an application layer configuration plugin.

Definition at line [96](#) of file [callback.doc](#).

6.17.2.5 void halBulbPwmDriverBlinkStartCallback (void)

This callback is generated when the application layer makes a call to initiate blinking behavior. It warns the application layer PWM code to not attempt to drive the LEDs directly and interfere with the blinking behavior.

Application Usage:

Should be implemented by an application layer configuration plugin.

Definition at line [124](#) of file [callback.doc](#).

6.17.2.6 void halBulbPwmDriverBlinkStopCallback (void)

This callback is generated when the current blinking command finishes. The application layer PWM code must then determine what the bulb drive should be, based on the current application layer attributes (i.e. level, on/off, color XY, etc.)

Application Usage:

Should be implemented by an application layer configuration plugin.

Definition at line [137](#) of file [callback.doc](#).

6.18 Button Callbacks

Functions

- void [halButtonIsr](#) (uint8_t button, uint8_t state)

6.18.1 Detailed Description

These callbacks are contributed by the Button plugin.

6.18.2 Function Documentation

6.18.2.1 void halButtonIsr (uint8_t *button*, uint8_t *state*)

A callback called in interrupt context whenever a button changes its state.

Application Usage:

Must be implemented by the application. This function should contain the functionality to be executed in response to changes of state in each of the buttons, or callbacks to the appropriate functionality.

Parameters

<i>button</i>	The button which has changed state, either BUTTON0 or BUTTON1 as defined in the appropriate BOARD_HEADER.
<i>state</i>	The new state of the button referenced by the button parameter, either ::BUTTON_PRESSED if the button has been pressed or ::BUTTON_RELEASED if the button has been released.

Definition at line 166 of file [callback.doc](#).

6.19 Debug Basic Library Callbacks

Functions

- void `emberDebugHandler` (EmberMessageBuffer message)

6.19.1 Detailed Description

These callbacks are contributed by the Debug Basic Library plugin.

6.19.2 Function Documentation

6.19.2.1 void `emberDebugHandler` (EmberMessageBuffer *message*)

Definition at line 181 of file [callback.doc](#).

6.20 EZSP Common Callbacks

Functions

- void `ezspErrorHandler` (EzspStatus status)
- void `ezspWaitingForResponse` (void)
- void `ezspNoCallbacks` (void)
- void `ezspStackTokenChangedHandler` (uint16_t tokenAddress)
- void `ezspTimerHandler` (uint8_t timerId)
- void `ezspCounterRolloverHandler` (EmberCounterType type)
- void `ezspCustomFrameHandler` (uint8_t payloadLength, uint8_t *payload)
- void `ezspStackStatusHandler` (EmberStatus status)
- void `ezspEnergyScanResultHandler` (uint8_t channel, int8_t maxRssiValue)
- void `ezspNetworkFoundHandler` (EmberZigbeeNetwork *networkFound, uint8_t lastHopLqi, int8_t lastHopRssi)
- void `ezspScanCompleteHandler` (uint8_t channel, EmberStatus status)
- void `ezspChildJoinHandler` (uint8_t index, bool joining, EmberNodeId childId, EmberEUI64 childEui64, EmberNodeType childType)
- void `ezspRemoteSetBindingHandler` (EmberBindingTableEntry *entry, uint8_t index, EmberStatus policyDecision)
- void `ezspRemoteDeleteBindingHandler` (uint8_t index, EmberStatus policyDecision)
- void `ezspMessageSentHandler` (EmberOutgoingMessageType type, uint16_t indexOrDestination, EmberApsFrame *apsFrame, uint8_t messageTag, EmberStatus status, uint8_t messageLength, uint8_t *messageContents)
- void `ezspPollCompleteHandler` (EmberStatus status)
- void `ezspPollHandler` (EmberNodeId childId)
- void `ezspIncomingSenderEui64Handler` (EmberEUI64 senderEui64)
- void `ezspIncomingMessageHandler` (EmberIncomingMessageType type, EmberApsFrame *apsFrame, uint8_t lastHopLqi, int8_t lastHopRssi, EmberNodeId sender, uint8_t bindingIndex, uint8_t addressIndex, uint8_t messageLength, uint8_t *messageContents)
- void `ezspIncomingRouteRecordHandler` (EmberNodeId source, EmberEUI64 sourceEui, uint8_t lastHopLqi, int8_t lastHopRssi, uint8_t relayCount, uint8_t *relayList)
- void `ezspIncomingManyToOneRouteRequestHandler` (EmberNodeId source, EmberEUI64 longId, uint8_t cost)
- void `ezspIncomingRouteErrorHandler` (EmberStatus status, EmberNodeId target)
- void `ezspIdConflictHandler` (EmberNodeId id)
- void `ezspMacPassthroughMessageHandler` (EmberMacPassthroughType messageType, uint8_t lastHopLqi, int8_t lastHopRssi, uint8_t messageLength, uint8_t *messageContents)
- void `ezspMacFilterMatchMessageHandler` (uint8_t filterIndexMatch, EmberMacPassthroughType legacyPassthroughType, uint8_t lastHopLqi, int8_t lastHopRssi, uint8_t messageLength, uint8_t *messageContents)
- void `ezspRawTransmitCompleteHandler` (EmberStatus status)
- void `ezspSwitchNetworkKeyHandler` (uint8_t sequenceNumber)
- void `ezspZigbeeKeyEstablishmentHandler` (EmberEUI64 partner, EmberKeyStatus status)
- void `ezspTrustCenterJoinHandler` (EmberNodeId newNodeId, EmberEUI64 newNodeEui64, EmberDeviceUpdate status, EmberJoinDecision policyDecision, EmberNodeId parentOfNewNodeId)
- void `ezspGenerateCbkeKeysHandler` (EmberStatus status, EmberPublicKeyData *ephemeralPublicKey)
- void `ezspCalculateSmacsHandler` (EmberStatus status, EmberSmacData *initiatorSmac, EmberSmacData *responderSmac)

- void `ezspGenerateCbkeKeysHandler283k1` (EmberStatus status, EmberPublicKey283k1Data *ephemeral-PublicKey)
- void `ezspCalculateSmacsHandler283k1` (EmberStatus status, EmberSmacData *initiatorSmac, Ember-SmacData *responderSmac)
- void `ezspDsaSignHandler` (EmberStatus status, uint8_t messageLength, uint8_t *messageContents)
- void `ezspDsaVerifyHandler` (EmberStatus status)
- void `ezspMfglibRxHandler` (uint8_t linkQuality, int8_t rssi, uint8_t packetLength, uint8_t *packet-Contents)
- void `ezspIncomingBootloadMessageHandler` (EmberEUI64 longId, uint8_t lastHopLqi, int8_t last-HopRssi, uint8_t messageLength, uint8_t *messageContents)
- void `ezspBootloadTransmitCompleteHandler` (EmberStatus status, uint8_t messageLength, uint8_t *messageContents)
- void `ezspZllNetworkFoundHandler` (EmberZllNetwork *networkInfo, bool isDeviceInfoNull, Ember-ZllDeviceInfoRecord *deviceInfo, uint8_t lastHopLqi, int8_t lastHopRssi)
- void `ezspZllScanCompleteHandler` (EmberStatus status)
- void `ezspZllAddressAssignmentHandler` (EmberZllAddressAssignment *addressInfo, uint8_t last-HopLqi, int8_t lastHopRssi)
- void `ezspZllTouchLinkTargetHandler` (EmberZllNetwork *networkInfo)
- void `ezspRf4ceIncomingMessageHandler` (uint8_t pairingIndex, uint8_t profileId, uint16_t vendor-Id, EmberRf4ceTxOption txOptions, uint8_t messageLength, uint8_t *message)
- void `ezspRf4ceMessageSentHandler` (EmberStatus status, uint8_t pairingIndex, EmberRf4ceTxOption txOptions, uint8_t profileId, uint16_t vendorId, uint8_t messageTag, uint8_t messageLength, uint8_t *message)
- void `ezspRf4ceDiscoveryCompleteHandler` (EmberStatus status)
- void `ezspRf4ceDiscoveryRequestHandler` (EmberEUI64 ieeeAddr, uint8_t nodeCapabilities, Ember-Rf4ceVendorInfo *vendorInfo, EmberRf4ceApplicationInfo *appInfo, uint8_t searchDevType, uint8-_t rxLinkQuality)
- void `ezspRf4ceDiscoveryResponseHandler` (bool atCapacity, uint8_t channel, EmberPanId panId, EmberEUI64 ieeeAddr, uint8_t nodeCapabilities, EmberRf4ceVendorInfo *vendorInfo, EmberRf4ce-ApplicationInfo *appInfo, uint8_t rxLinkQuality, uint8_t discRequestLqi)
- void `ezspRf4ceAutoDiscoveryResponseCompleteHandler` (EmberStatus status, EmberEUI64 src-IeeeAddr, uint8_t nodeCapabilities, EmberRf4ceVendorInfo *vendorInfo, EmberRf4ceApplication-Info *appInfo, uint8_t searchDevType)
- void `ezspRf4cePairCompleteHandler` (EmberStatus status, uint8_t pairingIndex, EmberRf4ceVendor-Info *vendorInfo, EmberRf4ceApplicationInfo *appInfo)
- void `ezspRf4cePairRequestHandler` (EmberStatus status, uint8_t pairingIndex, EmberEUI64 src-IeeeAddr, uint8_t nodeCapabilities, EmberRf4ceVendorInfo *vendorInfo, EmberRf4ceApplication-Info *appInfo, uint8_t keyExchangeTransferCount)
- void `ezspRf4ceUnpairHandler` (uint8_t pairingIndex)
- void `ezspRf4ceUnpairCompleteHandler` (uint8_t pairingIndex)
- void `ezspDGpSentHandler` (EmberStatus status, uint8_t gpepHandle)
- void `ezspGpepIncomingMessageHandler` (EmberStatus status, uint8_t gpdLink, uint8_t sequence-Number, EmberGpAddress *addr, EmberGpSecurityLevel gpdfSecurityLevel, EmberGpKeyType gpdfSecurityKeyType, bool autoCommissioning, bool rxAfterTx, uint32_t gpdSecurityFrameCounter, uint8_t gpdCommandId, uint32_t mic, EmberGpSinkListEntry *sinkList, uint8_t gpdCommand-PayloadLength, uint8_t *gpdCommandPayload)

6.20.1 Detailed Description

These callbacks are contributed by the EZSP Common plugin.

6.20.2 Function Documentation

6.20.2.1 void ezspErrorHandler (EzspStatus status)

Definition at line 200 of file [callback.doc](#).

6.20.2.2 void ezspWaitingForResponse (void)

Definition at line 209 of file [callback.doc](#).

6.20.2.3 void ezspNoCallbacks (void)

Definition at line 214 of file [callback.doc](#).

6.20.2.4 void ezspStackTokenChangedHandler (uint16_t tokenAddress)

Definition at line 219 of file [callback.doc](#).

6.20.2.5 void ezspTimerHandler (uint8_t timerId)

Definition at line 226 of file [callback.doc](#).

6.20.2.6 void ezspCounterRolloverHandler (EmberCounterType type)

Definition at line 233 of file [callback.doc](#).

6.20.2.7 void ezspCustomFrameHandler (uint8_t payloadLength, uint8_t * payload)

Definition at line 240 of file [callback.doc](#).

6.20.2.8 void ezspStackStatusHandler (EmberStatus status)

Definition at line 253 of file [callback.doc](#).

6.20.2.9 void ezspEnergyScanResultHandler (uint8_t channel, int8_t maxRssiValue)

Definition at line 262 of file [callback.doc](#).

6.20.2.10 void ezspNetworkFoundHandler (EmberZigbeeNetwork * networkFound, uint8_t lastHopLqi, int8_t lastHopRssi)

Definition at line 272 of file [callback.doc](#).

6.20.2.11 void ezspScanCompleteHandler (uint8_t channel, EmberStatus status)

Definition at line 285 of file [callback.doc](#).

6.20.2.12 void ezspChildJoinHandler (uint8_t *index*, bool *joining*, EmberNodeID *childId*, EmberEUI64 *childEui64*, EmberNodeType *childType*)

Definition at line 296 of file [callback.doc](#).

6.20.2.13 void ezspRemoteSetBindingHandler (EmberBindingTableEntry * *entry*, uint8_t *index*, EmberStatus *policyDecision*)

Definition at line 314 of file [callback.doc](#).

6.20.2.14 void ezspRemoteDeleteBindingHandler (uint8_t *index*, EmberStatus *policyDecision*)

Definition at line 329 of file [callback.doc](#).

6.20.2.15 void ezspMessageSentHandler (EmberOutgoingMessageType *type*, uint16_t *indexOrDestination*, EmberApsFrame * *apsFrame*, uint8_t *messageTag*, EmberStatus *status*, uint8_t *messageLength*, uint8_t * *messageContents*)

Definition at line 339 of file [callback.doc](#).

6.20.2.16 void ezspPollCompleteHandler (EmberStatus *status*)

Definition at line 364 of file [callback.doc](#).

6.20.2.17 void ezspPollHandler (EmberNodeID *childId*)

Definition at line 375 of file [callback.doc](#).

6.20.2.18 void ezspIncomingSenderEui64Handler (EmberEUI64 *senderEui64*)

Definition at line 385 of file [callback.doc](#).

6.20.2.19 void ezspIncomingMessageHandler (EmberIncomingMessageType *type*, EmberApsFrame * *apsFrame*, uint8_t *lastHopLqi*, int8_t *lastHopRssi*, EmberNodeID *sender*, uint8_t *bindingIndex*, uint8_t *addressIndex*, uint8_t *messageLength*, uint8_t * *messageContents*)

Definition at line 392 of file [callback.doc](#).

6.20.2.20 void ezspIncomingRouteRecordHandler (EmberNodeID *source*, EmberEUI64 *sourceEui*, uint8_t *lastHopLqi*, int8_t *lastHopRssi*, uint8_t *relayCount*, uint8_t * *relayList*)

Definition at line 420 of file [callback.doc](#).

6.20.2.21 void ezspIncomingManyToOneRouteRequestHandler (EmberNodeID *source*, EmberEUI64 *longId*, uint8_t *cost*)

Definition at line 439 of file [callback.doc](#).

6.20.2.22 void ezspIncomingRouteErrorHandler (EmberStatus *status*, EmberNodeid *target*)

Definition at line [453](#) of file [callback.doc](#).

6.20.2.23 void ezspIdConflictHandler (EmberNodeid *id*)

Definition at line [466](#) of file [callback.doc](#).

6.20.2.24 void ezspMacPassthroughMessageHandler (EmberMacPassthroughType *messageType*, uint8_t *lastHopLqi*, int8_t *lastHopRssi*, uint8_t *messageLength*, uint8_t * *messageContents*)

Definition at line [474](#) of file [callback.doc](#).

6.20.2.25 void ezspMacFilterMatchMessageHandler (uint8_t *filterIndexMatch*, EmberMacPassthroughType *legacyPassthroughType*, uint8_t *lastHopLqi*, int8_t *lastHopRssi*, uint8_t *messageLength*, uint8_t * *messageContents*)

Definition at line [490](#) of file [callback.doc](#).

6.20.2.26 void ezspRawTransmitCompleteHandler (EmberStatus *status*)

Definition at line [508](#) of file [callback.doc](#).

6.20.2.27 void ezspSwitchNetworkKeyHandler (uint8_t *sequenceNumber*)

Definition at line [518](#) of file [callback.doc](#).

6.20.2.28 void ezspZigbeeKeyEstablishmentHandler (EmberEUI64 *partner*, EmberKeyStatus *status*)

Definition at line [526](#) of file [callback.doc](#).

6.20.2.29 void ezspTrustCenterJoinHandler (EmberNodeid *newNodeId*, EmberEUI64 *newNodeEui64*, EmberDeviceUpdate *status*, EmberJoinDecision *policyDecision*, EmberNodeid *parentOfNewNodeId*)

Definition at line [540](#) of file [callback.doc](#).

6.20.2.30 void ezspGenerateCbkeKeysHandler (EmberStatus *status*, EmberPublicKeyData * *ephemeralPublicKey*)

Definition at line [559](#) of file [callback.doc](#).

6.20.2.31 void ezspCalculateSmacsHandler (EmberStatus *status*, EmberSmacData * *initiatorSmac*, EmberSmacData * *responderSmac*)

Definition at line [571](#) of file [callback.doc](#).

6.20.2.32 void ezspGenerateCbkeKeysHandler283k1 (EmberStatus *status*, EmberPublicKey283k1Data * *ephemeralPublicKey*)

Definition at line [585](#) of file [callback.doc](#).

6.20.2.33 void ezspCalculateSmacsHandler283k1 (EmberStatus *status*, EmberSmacData * *initiatorSmac*, EmberSmacData * *responderSmac*)

Definition at line [598](#) of file [callback.doc](#).

6.20.2.34 void ezspDsaSignHandler (EmberStatus *status*, uint8_t *messageLength*, uint8_t * *messageContents*)

Definition at line [612](#) of file [callback.doc](#).

6.20.2.35 void ezspDsaVerifyHandler (EmberStatus *status*)

Definition at line [628](#) of file [callback.doc](#).

6.20.2.36 void ezspMfglibRxHandler (uint8_t *linkQuality*, int8_t *rssi*, uint8_t *packetLength*, uint8_t * *packetContents*)

Definition at line [635](#) of file [callback.doc](#).

6.20.2.37 void ezspIncomingBootloadMessageHandler (EmberEUI64 *longId*, uint8_t *lastHopLqi*, int8_t *lastHopRssi*, uint8_t *messageLength*, uint8_t * *messageContents*)

Definition at line [650](#) of file [callback.doc](#).

6.20.2.38 void ezspBootloadTransmitCompleteHandler (EmberStatus *status*, uint8_t *messageLength*, uint8_t * *messageContents*)

Definition at line [666](#) of file [callback.doc](#).

6.20.2.39 void ezspZllNetworkFoundHandler (EmberZllNetwork * *networkInfo*, bool *isDeviceInfoNull*, EmberZllDeviceInfoRecord * *deviceInfo*, uint8_t *lastHopLqi*, int8_t *lastHopRssi*)

Definition at line [678](#) of file [callback.doc](#).

6.20.2.40 void ezspZllScanCompleteHandler (EmberStatus *status*)

Definition at line [693](#) of file [callback.doc](#).

6.20.2.41 void ezspZllAddressAssignmentHandler (EmberZllAddressAssignment * *addressInfo*, uint8_t *lastHopLqi*, int8_t *lastHopRssi*)

Definition at line [701](#) of file [callback.doc](#).

6.20.2.42 void ezspZllTouchLinkTargetHandler (EmberZllNetwork * *networkInfo*)

Definition at line 712 of file [callback.doc](#).

6.20.2.43 void ezspRf4ceIncomingMessageHandler (uint8_t *pairingIndex*, uint8_t *profileId*, uint16_t *vendorId*, EmberRf4ceTxOption *txOptions*, uint8_t *messageLength*, uint8_t * *message*)

Definition at line 719 of file [callback.doc](#).

6.20.2.44 void ezspRf4ceMessageSentHandler (EmberStatus *status*, uint8_t *pairingIndex*, EmberRf4ceTxOption *txOptions*, uint8_t *profileId*, uint16_t *vendorId*, uint8_t *messageTag*, uint8_t *messageLength*, uint8_t * *message*)

Definition at line 739 of file [callback.doc](#).

6.20.2.45 void ezspRf4ceDiscoveryCompleteHandler (EmberStatus *status*)

Definition at line 765 of file [callback.doc](#).

6.20.2.46 void ezspRf4ceDiscoveryRequestHandler (EmberEUI64 *ieeeAddr*, uint8_t *nodeCapabilities*, EmberRf4ceVendorInfo * *vendorInfo*, EmberRf4ceApplicationInfo * *applInfo*, uint8_t *searchDevType*, uint8_t *rxLinkQuality*)

Definition at line 778 of file [callback.doc](#).

6.20.2.47 void ezspRf4ceDiscoveryResponseHandler (bool *atCapacity*, uint8_t *channel*, EmberPanId *panId*, EmberEUI64 *ieeeAddr*, uint8_t *nodeCapabilities*, EmberRf4ceVendorInfo * *vendorInfo*, EmberRf4ceApplicationInfo * *applInfo*, uint8_t *rxLinkQuality*, uint8_t *discRequestLqi*)

Definition at line 801 of file [callback.doc](#).

6.20.2.48 void ezspRf4ceAutoDiscoveryResponseCompleteHandler (EmberStatus *status*, EmberEUI64 *srcIEEEAddr*, uint8_t *nodeCapabilities*, EmberRf4ceVendorInfo * *vendorInfo*, EmberRf4ceApplicationInfo * *applInfo*, uint8_t *searchDevType*)

Definition at line 828 of file [callback.doc](#).

6.20.2.49 void ezspRf4cePairCompleteHandler (EmberStatus *status*, uint8_t *pairingIndex*, EmberRf4ceVendorInfo * *vendorInfo*, EmberRf4ceApplicationInfo * *applInfo*)

Definition at line 856 of file [callback.doc](#).

6.20.2.50 void ezspRf4cePairRequestHandler (EmberStatus *status*, uint8_t *pairingIndex*, EmberEUI64 *srcIEEEAddr*, uint8_t *nodeCapabilities*, EmberRf4ceVendorInfo * *vendorInfo*, EmberRf4ceApplicationInfo * *applInfo*, uint8_t *keyExchangeTransferCount*)

Definition at line 874 of file [callback.doc](#).

6.20.2.51 void ezspRf4ceUnpairHandler (*uint8_t pairingIndex*)

Definition at line [897](#) of file [callback.doc](#).

6.20.2.52 void ezspRf4ceUnpairCompleteHandler (*uint8_t pairingIndex*)

Definition at line [905](#) of file [callback.doc](#).

6.20.2.53 void ezspDGpSentHandler (*EmberStatus status, uint8_t gpephandle*)

Definition at line [913](#) of file [callback.doc](#).

6.20.2.54 void ezspGpepIncomingMessageHandler (*EmberStatus status, uint8_t gpdLink, uint8_t sequenceNumber, EmberGpAddress * addr, EmberGpSecurityLevel gpdfSecurityLevel, EmberGpKeyType gpdfSecurityKeyType, bool autoCommissioning, bool rxAfterTx, uint32_t gpdSecurityFrameCounter, uint8_t gpdCommandId, uint32_t mic, EmberGpSinkListEntry * sinkList, uint8_t gpdCommandPayloadLength, uint8_t * gpdCommandPayload*)

Definition at line [922](#) of file [callback.doc](#).

6.21 HAL Library Callbacks

Functions

- void [halRadioPowerUpHandler](#) (void)
- void [halRadioPowerDownHandler](#) (void)

6.21.1 Detailed Description

These callbacks are contributed by the HAL Library plugin.

6.21.2 Function Documentation

6.21.2.1 void [halRadioPowerUpHandler](#) (void)

Handler called whenever the radio is powered on.

Definition at line [967](#) of file [callback.doc](#).

6.21.2.2 void [halRadioPowerDownHandler](#) (void)

Handler called whenever the radio is powered off.

Definition at line [974](#) of file [callback.doc](#).

6.22 Idle/Sleep Callbacks

Functions

- bool `emberAfPluginIdleSleepOkToSleepCallback` (`uint32_t durationMs`)
- void `emberAfPluginIdleSleepWakeUpCallback` (`uint32_t durationMs`)
- bool `emberAfPluginIdleSleepOkToIdleCallback` (`void`)
- void `emberAfPluginIdleSleepActiveCallback` (`void`)

6.22.1 Detailed Description

These callbacks are contributed by the Idle/Sleep plugin.

6.22.2 Function Documentation

6.22.2.1 bool `emberAfPluginIdleSleepOkToSleepCallback` (`uint32_t durationMs`)

Ok To Sleep.

This function is called by the Idle/Sleep plugin before sleeping. It is called with interrupts disabled. The application should return true if the device may sleep or false otherwise.

Parameters

<code>durationMs</code>	The maximum duration in milliseconds that the device will sleep. Ver.: always
-------------------------	---

Definition at line 998 of file `callback.doc`.

6.22.2.2 void `emberAfPluginIdleSleepWakeUpCallback` (`uint32_t durationMs`)

Wake Up.

This function is called by the Idle/Sleep plugin after sleeping.

Parameters

<code>durationMs</code>	The duration in milliseconds that the device slept. Ver.: always
-------------------------	--

Definition at line 1010 of file `callback.doc`.

6.22.2.3 bool `emberAfPluginIdleSleepOkToIdleCallback` (`void`)

Ok To Idle.

This function is called by the Idle/Sleep plugin before idling. It is called with interrupts disabled. The application should return true if the device may idle or false otherwise.

Definition at line 1021 of file `callback.doc`.

6.22.2.4 void `emberAfPluginIdleSleepActiveCallback` (`void`)

Active.

This function is called by the Idle/Sleep plugin after idling.

Definition at line 1031 of file [callback.doc](#).

6.23 Low Voltage Shutdown Callbacks

Functions

- bool `emberAfPluginLowVoltageShutdownOkToShutdownCallback` (`uint16_t shutdownVoltage`)
- void `emberAfPluginLowVoltageShutdownPreShutdownCallback` (`uint16_t shutdownVoltage`)

6.23.1 Detailed Description

These callbacks are contributed by the Low Voltage Shutdown plugin.

6.23.2 Function Documentation

6.23.2.1 bool `emberAfPluginLowVoltageShutdownOkToShutdownCallback` (`uint16_t shutdownVoltage`)

Ok To Shutdown.

This function is called by the Low Voltage Shutdown plugin after the low voltage threshold has been detected but before shutting down the chip (until a full power-on reset is triggered). The application should return true if the device may shutdown or false otherwise. Note that this callback is not the proper place to put logic to be performed just prior to shutdown; that would be the Pre Shutdown Callback. If callback returns false, shutdown check will be performed again in EMBER_AF_PLUGIN_LOW_VOLTAGE_SHUTDOWN_POSTPONE_DURATION_MS milliseconds.

Parameters

<code>shutdown-Voltage</code>	The voltage (in mV) read from VDD at shutdown time, which can be used to determine the risk in doing any further activities prior to shutdown. Ver.: always
-------------------------------	---

Definition at line 1061 of file [callback.doc](#).

6.23.2.2 void `emberAfPluginLowVoltageShutdownPreShutdownCallback` (`uint16_t shutdownVoltage`)

Pre Shutdown.

This function is called by the Low Voltage Shutdown plugin just prior to shutting down, allowing the application to do any last-minute clean-up before the chip goes offline until next reset. (Can't prevent shutdown from this context; use Ok To Shutdown callback for that instead.)

Parameters

<code>shutdown-Voltage</code>	The voltage (in mV) read from VDD at shutdown time, which can be used to determine the risk in doing any further activities prior to shutdown. Ver.: always
-------------------------------	---

Definition at line 1077 of file [callback.doc](#).

6.24 Main Callbacks

Functions

- void `emberAfMainInitCallback` (void)
- void `emberAfMainTickCallback` (void)
- void `emberAfStackStatusCallback` (EmberStatus status)
- void `emberAfNcpInitCallback` (bool memoryAllocation)

6.24.1 Detailed Description

These callbacks are contributed by the Main plugin.

6.24.2 Function Documentation

6.24.2.1 void `emberAfMainInitCallback` (void)

Main Init.

This function is called when the application starts and can be used to perform any additional initialization required at system startup.

Definition at line 1097 of file [callback.doc](#).

6.24.2.2 void `emberAfMainTickCallback` (void)

Main Tick.

This function is called in each iteration of the main application loop and can be used to perform periodic functions. The frequency with which this function is called depends on how quickly the main loop runs. If the application blocks at any time during the main loop, this function will not be called until execution resumes. On SoC platforms, sleeping and idling will block. On Unix hosts, process yielding (e.g., via select) will block.

Definition at line 1110 of file [callback.doc](#).

6.24.2.3 void `emberAfStackStatusCallback` (EmberStatus status)

Stack Status.

This function is called when the stack status changes. This callbacks provides applications an opportunity to be notified of changes to the stack status and take appropriate action.

Parameters

<code>status</code>	Ver.: always
---------------------	--------------

Definition at line 1122 of file [callback.doc](#).

6.24.2.4 void `emberAfNcpInitCallback` (bool *memoryAllocation*)

Ncp Init.

This function is called when the network coprocessor is being initialized, either at startup or upon reset. It provides applications an opportunity to perform additional configuration of the NCP. The function is always called twice when the NCP is initialized. In the first invocation, memoryAllocation will be true and the application should only issue EZSP commands that affect memory allocation on the NCP. For example, tables on the NCP can be resized in the first call. In the second invocation, memoryAllocation will be false and the application should only issue EZSP commands that do not affect memory allocation. For example, tables on the NCP can be populated in the second call. This callback is not called on SoCs.

Parameters

<i>memory- Allocation</i>	Ver.: always
-------------------------------	--------------

Definition at line 1141 of file [callback.doc](#).

6.25 Microphone Codec MSADPCM Callbacks

Functions

- void `halMicrophoneCodecMsadpcmDataReadyCallback` (`uint8_t *data, uint8_t length`)

6.25.1 Detailed Description

These callbacks are contributed by the Microphone Codec MSADPCM plugin.

6.25.2 Function Documentation

6.25.2.1 void `halMicrophoneCodecMsadpcmDataReadyCallback` (`uint8_t * data, uint8_t length`)

A callback called when new microphone data is ready.

Application Usage:

This function is called by the plugin when new data has been processed and is ready to be processed by other parts of the system.

Parameters

<i>data</i>	Pointer to the data that is ready
<i>length</i>	Length of the data

Definition at line 1165 of file [callback.doc](#).

6.26 Microphone IMAADPCM Callbacks

Functions

- void [halMicrophoneImaadpcmDataReadyCallback](#) (uint8_t *data, uint8_t length)

6.26.1 Detailed Description

These callbacks are contributed by the Microphone IMAADPCM plugin.

6.26.2 Function Documentation

6.26.2.1 void [halMicrophonelmaadpcmDataReadyCallback](#) (uint8_t * *data*, uint8_t *length*)

A callback called when new microphone data is ready.

Application Usage:

This function is called by the plugin when new data has been processed and is ready to be processed by other parts of the system.

Parameters

<i>data</i>	Pointer to the data that is ready
<i>length</i>	Length of the data

Definition at line [1189](#) of file [callback.doc](#).

6.27 Generic Device Profile Callbacks

Functions

- bool `emberAfPluginRf4ceGdpZrc20StartConfigurationCallback` (bool `isOriginator`, uint8_t `pairingIndex`)
- void `emberAfPluginRf4ceGdpZrc20BindingCompleteCallback` (`EmberAfRf4ceGdpBindingStatus` `status`, uint8_t `pairingIndex`)
- void `emberAfPluginRf4ceGdpStartValidationCallback` (uint8_t `pairingIndex`)
- void `emberAfPluginRf4ceGdpBindingCompleteCallback` (`EmberAfRf4ceGdpBindingStatus` `status`, uint8_t `pairingIndex`)
- bool `emberAfPluginRf4ceGdpIncomingBindProxyCallback` (const `EmberEUI64` `ieeeAddr`)
- void `emberAfPluginRf4ceGdpHeartbeatPollingEstablishedCallback` (uint8_t `pairingIndex`, `EmberAfRf4ceGdpPollingTrigger` `triggers`)
- void `emberAfPluginRf4ceGdpIdentifyCallback` (`EmberAfRf4ceGdpClientNotificationIdentifyFlags` `flags`, uint16_t `timeS`)
- void `emberAfPluginRf4ceGdpIdentifyClientFoundCallback` (`EmberAfRf4ceGdpClientNotificationIdentifyFlags` `flags`)
- void `emberAfPluginRf4ceGdpKeyExchangeCompleteCallback` (`EmberStatus` `status`)
- bool `emberAfPluginRf4ceGdpVendorSpecificKeyExchangeCallback` (uint8_t `initiatorVendorSpecificParam`, uint8_t *`responderVendorSpecificParam`, uint8_t *`sharedSecret`)

6.27.1 Detailed Description

These callbacks are contributed by the Generic Device Profile plugin.

6.27.2 Function Documentation

6.27.2.1 bool `emberAfPluginRf4ceGdpZrc20StartConfigurationCallback` (`bool isOriginator`, `uint8_t pairingIndex`)

Zrc 2 0 Start Configuration.

This function is called by the RF4CE GDP plugin when ZRC 2.0 configuration should begin.

Parameters

<code>isOriginator</code>	true if is originator, false if is recipient. Ver.: always
<code>pairingIndex</code>	The index of the pairing entry. Ver.: always

Definition at line 1213 of file `callback.doc`.

6.27.2.2 void `emberAfPluginRf4ceGdpZrc20BindingCompleteCallback` (`EmberAfRf4ceGdpBindingStatus status`, `uint8_t pairingIndex`)

Zrc 2 0 Binding Complete.

This function is called by the RF4CE GDP plugin when a ZRC 2.0 binding procedure completed.

Parameters

<i>status</i>	An <code>EmberAfRf4ceGdpBindingStatus</code> value indicating whether the binding succeeded or the reason of failure. Ver.: always
<i>pairingIndex</i>	The index of the pairing entry. Ver.: always

Definition at line 1228 of file `callback.doc`.

6.27.2.3 void emberAfPluginRf4ceGdpStartValidationCallback (uint8_t pairingIndex)

Start Validation.

This function is called by the RF4CE GDP plugin when the application should begin the validation procedure. The application must complete the validation within the validation wait time to avoid the validation automatically failing due to a timeout.

Parameters

<i>pairingIndex</i>	The index of the pairing entry. Ver.: always
---------------------	--

Definition at line 1242 of file `callback.doc`.

6.27.2.4 void emberAfPluginRf4ceGdpBindingCompleteCallback (EmberAfRf4ceGdpBindingStatus status, uint8_t pairingIndex)

Binding Complete.

This function is called by the RF4CE GDP plugin when the binding operation completes. If status is ::EMBER_SUCCESS, binding was successful and pairingIndex indicates the index in the pairing table for the remote node.

Parameters

<i>status</i>	The status of the binding operation. Ver.: always
<i>pairingIndex</i>	The index of the pairing entry. Ver.: always

Definition at line 1255 of file `callback.doc`.

6.27.2.5 bool emberAfPluginRf4ceGdpIncomingBindProxyCallback (const EmberEUI64 ieeeAddr)

Incoming Bind Proxy.

This function is called by the RF4CE GDP plugin when the binding recipient receives a pair request with the Binding Proxy Supported bit set. If this callback returns true, the recipient will respond to the pair request, otherwise it will ignore it.

Parameters

<i>ieeeAddr</i>	The IEEE address of the originator node sending the pair request. Ver.: always
-----------------	--

Definition at line 1270 of file `callback.doc`.

6.27.2.6 void emberAfPluginRf4ceGdpHeartbeatPollingEstablishedCallback (*uint8_t pairingIndex*, *EmberAfRf4ceGdpPollingTrigger triggers*)

Heartbeat Polling Established.

This function is called by the RF4CE GDP plugin when heartbeat polling is established between a client and this server. The application should wait until the heartbeat callback is called to send messages to the client. A module can subscribe to incoming heartbeats by using the [emberAfRf4ceGdpSubscribeToHeartbeat\(\)](#) API.

Parameters

<i>pairingIndex</i>	The polling method. Ver.: always
<i>triggers</i>	The polling triggers. Ver.: always

Definition at line [1286](#) of file [callback.doc](#).

6.27.2.7 void emberAfPluginRf4ceGdpIdentifyCallback (*EmberAfRf4ceGdpClientNotificationIdentifyFlags flags*, *uint16_t timeS*)

Identify.

This function is called by the RF4CE GDP plugin when the client should take an action to identify itself.

Parameters

<i>flags</i>	The actions the client should take to identify itself. Ver.: always
<i>timeS</i>	The time in seconds that the client should perform the identify action or zero if the client should stop the action. Ver.: always

Definition at line [1301](#) of file [callback.doc](#).

6.27.2.8 void emberAfPluginRf4ceGdpIdentifyClientFoundCallback (*EmberAfRf4ceGdpClientNotificationIdentifyFlags flags*)

Identify Client Found.

This function is called by the RF4CE GDP plugin when the an identify server discovers an identify client.

Parameters

<i>flags</i>	The actions supported by the discovered identify client. Ver.: always
--------------	---

Definition at line [1314](#) of file [callback.doc](#).

6.27.2.9 void emberAfPluginRf4ceGdpKeyExchangeCompleteCallback (*EmberStatus status*)

Key Exchange Complete.

This function is called by the RF4CE GDP plugin when a Key Exchange procedure that was initiated by the application has completed.

Parameters

<i>status</i>	The status success/failed of the Key Exchange procedure. Ver.: always
---------------	---

Definition at line 1326 of file [callback.doc](#).

6.27.2.10 bool emberAfPluginRf4ceGdpVendorSpecificKeyExchangeCallback (uint8_t initiatorVendorSpecificParam, uint8_t * responderVendorSpecificParam, uint8_t * sharedSecret)

Vendor Specific Key Exchange.

This function is called by the RF4CE GDP plugin when a Key Exchange Challenge with the vendor-specific bit set is received. If this callback returns true, the node will respond with a Key Exchange Challenge Response command with the vendor-specific bit set.

Parameters

<i>initiator-VendorSpecific-Param</i>	The additional parameter included in the Key Exchange Flags of the incoming Key Exchange Challenge command. Ver.: always
<i>responder-VendorSpecific-Param</i>	The additional parameter to be included in the Key Exchange Flags of the Key Exchange Challenge Response command the node will send out. Ver.: always
<i>sharedSecret</i>	A pointer to an 8-byte area memory to be filled with the vendor-specific shared secret. Ver.: always

Definition at line 1346 of file [callback.doc](#).

6.28 RF4CE Multiple System Operators Profile Callbacks

Functions

- void `emberAfPluginRf4ceMsoStartValidationCallback` (uint8_t pairingIndex)
- void `emberAfPluginRf4ceMsoBindingCompleteCallback` (EmberAfRf4ceMsoBindingStatus status, uint8_t pairingIndex)
- void `emberAfPluginRf4ceMsoUserControlCallback` (const EmberAfRf4ceMsoUserControlRecord *record)
- EmberAfRf4ceStatus `emberAfPluginRf4ceMsoGetIrRfDatabaseAttributeCallback` (uint8_t pairingIndex, uint8_t entryIndex, uint8_t *valueLength, uint8_t *value)
- bool `emberAfPluginRf4ceMsoHaveIrRfDatabaseAttributeCallback` (uint8_t pairingIndex, uint8_t entryIndex)
- void `emberAfPluginRf4ceMsoIncomingIrRfDatabaseAttributeCallback` (uint8_t pairingIndex, uint8_t entryIndex, uint8_t valueLength, const uint8_t *value)
- EmberStatus `emberAfPluginRf4ceMsoGetIrRfDatabaseEntryCallback` (EmberAfRf4ceMsoKeyCode keyCode, EmberAfRf4ceMsoIrRfDatabaseEntry *entry)

6.28.1 Detailed Description

These callbacks are contributed by the RF4CE Multiple System Operators Profile plugin.

6.28.2 Function Documentation

6.28.2.1 void `emberAfPluginRf4ceMsoStartValidationCallback` (uint8_t *pairingIndex*)

Start Validation.

This function is called by the RF4CE MSO plugin when a temporary pairing has been successfully created and the application should begin the validation procedure. The application must complete the validation within the validation wait time (if set) and also kick the watchdog within the initial watchdog timeout (if set) to avoid the validation automatically failing due to a timeout.

Parameters

<i>pairingIndex</i>	The index of the pairing entry. Ver.: always
---------------------	--

Definition at line 1376 of file `callback.doc`.

6.28.2.2 void `emberAfPluginRf4ceMsoBindingCompleteCallback` (EmberAfRf4ceMsoBindingStatus *status*, uint8_t *pairingIndex*)

Binding Complete.

This function is called by the RF4CE MSO plugin when the binding operation completes. If status is ::EMBER_SUCCESS, binding was successful and pairingIndex indicates the index in the pairing table for the remote node.

Parameters

<i>status</i>	The status of the binding operation. Ver.: always
<i>pairingIndex</i>	The index of the pairing entry. Ver.: always

Definition at line 1389 of file [callback.doc](#).

6.28.2.3 void emberAfPluginRf4ceMsoUserControlCallback (const EmberAfRf4ceMsoUserControlRecord * record)

User Control.

This function is called by the RF4CE MSO plugin when a user control starts or stops. If the type of the record is [EMBER_AF_RF4CE_MSO_COMMAND_USER_CONTROL_PRESSED](#), the application should execute the requested operation repeatedly at some application-specific rate. When the repetition should stop, the plugin will call the callback again with the type set to [EMBER_AF_RF4CE_MSO_COMMAND_USER_CONTROL_RELEASED](#). [EMBER_AF_RF4CE_MSO_COMMAND_USER_CONTROL_REPEAT](#) is a special case of [EMBER_AF_RF4CE_MSO_COMMAND_USER_CONTROL_PRESSED](#) and means that the pressed command from the originator was not received and that the originator is still triggering the action. The application should process a repeat type the same as a pressed type, but may wish to perform additional operations to compensate for missed commands.

Parameters

<i>record</i>	The user control record. Ver.: always
---------------	---------------------------------------

Definition at line 1411 of file [callback.doc](#).

6.28.2.4 EmberAfRf4ceStatus emberAfPluginRf4ceMsoGetIrRfDatabaseAttributeCallback (uint8_t pairingIndex, uint8_t entryIndex, uint8_t * valueLength, uint8_t * value)

Get Ir Rf Database Attribute.

This function is called by the RF4CE MSO plugin when a target is queried by a controller for one of the IR_RF_DATABASE attributes. The application is responsible for storing these attributes. If the application currently stores an entry for the corresponding passed pairing index and entry index, it should copy the content of such attribute at the passed 'value' pointer and return [EMBER_AF_RF4CE_STATUS_SUCCESS](#), otherwise it should return an [EmberAfRf4ceStatus](#) indicating the reason of failure. *valueLength* is a pointer to an integer that indicates the length of the buffer pointed to by *value*. The application must not write more than **valueLength* bytes to *value*. If the application returns [EMBER_AF_RF4CE_STATUS_SUCCESS](#), it must also update **valueLength* to indicate the number of bytes actually written to *value*.

Parameters

<i>pairingIndex</i>	The pairing index of the requested attribute. Ver.: always
<i>entryIndex</i>	The entry index of the requested attribute. Ver.: always
<i>valueLength</i>	A pointer to the size of the buffer pointed to by <i>value</i> that must be updated with the length written to <i>value</i> . Ver.: always
<i>value</i>	A pointer where the application should copy the content of the requested attribute. Ver.: always

Definition at line 1438 of file [callback.doc](#).

References [EMBER_AF_RF4CE_STATUS_UNSUPPORTED_ATTRIBUTE](#).

6.28.2.5 bool emberAfPluginRf4ceMsoHaveIrRfDatabaseAttributeCallback (*uint8_t pairingIndex, uint8_t entryIndex*)

Have Ir Rf Database Attribute.

This function is called by the RF4CE MSO plugin when a target is queried by a controller for one of the I_R_RF_DATABASE attributes. The application is responsible for storing these attributes. If the application currently stores an entry for the corresponding passed pairing index and entry index, it should return true, otherwise it should return false.

Parameters

<i>pairingIndex</i>	The pairing index of the requested attribute. Ver.: always
<i>entryIndex</i>	The entry index of the requested attribute. Ver.: always

Definition at line 1458 of file [callback.doc](#).

6.28.2.6 void emberAfPluginRf4ceMsolIncomingIrRfDatabaseAttributeCallback (*uint8_t pairingIndex, uint8_t entryIndex, uint8_t valueLength, const uint8_t * value*)

Incoming Ir Rf Database Attribute.

This function is called by the RF4CE MSO plugin when controller receives a get attribute response for one of the IR_RF_DATABASE attributes. The application is responsible for storing these attributes.

Parameters

<i>pairingIndex</i>	The pairing index of the requested attribute. Ver.: always
<i>entryIndex</i>	The entry index of the requested attribute. Ver.: always
<i>valueLength</i>	The length of the requested attribute. Ver.: always
<i>value</i>	A pointer to the content of the requested attribute. Ver.: always

Definition at line 1477 of file [callback.doc](#).

6.28.2.7 EmberStatus emberAfPluginRf4ceMsoGetIrRfDatabaseEntryCallback (*EmberAfRf4ceMsoKeyCode keyCode, EmberAfRf4ceMsoIrRfDatabaseEntry * entry*)

Get Ir Rf Database Entry.

This function is called by the RF4CE MSO plugin to retrieve the IR-RF database entry for a key code when the controller is sending user control commands. The application should return ::EMBER_SUCCESS and populate the entry if the IR-RF database entry is available for the RC command code. Otherwise, the application should return an error code.

Parameters

<i>keyCode</i>	The RC command code of the user control. Ver.: always
<i>entry</i>	A pointer to the EmberAfRf4ceMsoIrRfDatabaseEntry to be populated. Ver.: always

Definition at line 1496 of file [callback.doc](#).

6.29 RF4CE Profile Support Callbacks

Functions

- bool `emberAfPluginRf4ceProfileGdpMessageSentCallback` (uint8_t pairingIndex, uint8_t profileId, uint16_t vendorId, uint8_t messageTag, const uint8_t *message, uint8_t messageLength, EmberStatus status)
- void `emberAfPluginRf4ceProfileRemoteControl11MessageSentCallback` (uint8_t pairingIndex, uint16_t vendorId, uint8_t messageTag, const uint8_t *message, uint8_t messageLength, EmberStatus status)
- void `emberAfPluginRf4ceProfileZrc20MessageSentCallback` (uint8_t pairingIndex, uint16_t vendorId, uint8_t messageTag, const uint8_t *message, uint8_t messageLength, EmberStatus status)
- void `emberAfPluginRf4ceProfileMsoMessageSentCallback` (uint8_t pairingIndex, uint16_t vendorId, uint8_t messageTag, const uint8_t *message, uint8_t messageLength, EmberStatus status)
- void `emberAfPluginRf4ceProfileMessageSentCallback` (uint8_t pairingIndex, uint8_t profileId, uint16_t vendorId, uint8_t messageTag, const uint8_t *message, uint8_t messageLength, EmberStatus status)
- bool `emberAfPluginRf4ceProfileGdpIncomingMessageCallback` (uint8_t pairingIndex, uint8_t profileId, uint16_t vendorId, EmberRf4ceTxOption txOptions, const uint8_t *message, uint8_t messageLength)
- void `emberAfPluginRf4ceProfileRemoteControl11IncomingMessageCallback` (uint8_t pairingIndex, uint16_t vendorId, EmberRf4ceTxOption txOptions, const uint8_t *message, uint8_t messageLength)
- void `emberAfPluginRf4ceProfileZrc20IncomingMessageCallback` (uint8_t pairingIndex, uint16_t vendorId, EmberRf4ceTxOption txOptions, const uint8_t *message, uint8_t messageLength)
- void `emberAfPluginRf4ceProfileMsoIncomingMessageCallback` (uint8_t pairingIndex, uint16_t vendorId, EmberRf4ceTxOption txOptions, const uint8_t *message, uint8_t messageLength)
- void `emberAfPluginRf4ceProfileIncomingMessageCallback` (uint8_t pairingIndex, uint8_t profileId, uint16_t vendorId, EmberRf4ceTxOption txOptions, const uint8_t *message, uint8_t messageLength)
- bool `emberAfPluginRf4ceProfileGdpDiscoveryRequestCallback` (const EmberEUI64 ieeeAddr, uint8_t nodeCapabilities, const EmberRf4ceVendorInfo *vendorInfo, const EmberRf4ceApplicationInfo *appInfo, uint8_t searchDevType, uint8_t rxLinkQuality)
- bool `emberAfPluginRf4ceProfileRemoteControl11DiscoveryRequestCallback` (const EmberEUI64 ieeeAddr, uint8_t nodeCapabilities, const EmberRf4ceVendorInfo *vendorInfo, const EmberRf4ceApplicationInfo *appInfo, uint8_t searchDevType, uint8_t rxLinkQuality)
- bool `emberAfPluginRf4ceProfileZrc20DiscoveryRequestCallback` (const EmberEUI64 ieeeAddr, uint8_t nodeCapabilities, const EmberRf4ceVendorInfo *vendorInfo, const EmberRf4ceApplicationInfo *appInfo, uint8_t searchDevType, uint8_t rxLinkQuality)
- bool `emberAfPluginRf4ceProfileMsoDiscoveryRequestCallback` (const EmberEUI64 ieeeAddr, uint8_t nodeCapabilities, const EmberRf4ceVendorInfo *vendorInfo, const EmberRf4ceApplicationInfo *appInfo, uint8_t searchDevType, uint8_t rxLinkQuality)
- bool `emberAfPluginRf4ceProfileGdpDiscoveryResponseCallback` (bool atCapacity, uint8_t channel, EmberPanId panId, const EmberEUI64 ieeeAddr, uint8_t nodeCapabilities, const EmberRf4ceVendorInfo *vendorInfo, const EmberRf4ceApplicationInfo *appInfo, uint8_t rxLinkQuality, uint8_t discRequestLqi)
- bool `emberAfPluginRf4ceProfileRemoteControl11DiscoveryResponseCallback` (bool atCapacity, uint8_t channel, EmberPanId panId, const EmberEUI64 ieeeAddr, uint8_t nodeCapabilities, const EmberRf4ceVendorInfo *vendorInfo, const EmberRf4ceApplicationInfo *appInfo, uint8_t rxLinkQuality, uint8_t discRequestLqi)
- bool `emberAfPluginRf4ceProfileZrc20DiscoveryResponseCallback` (bool atCapacity, uint8_t channel, EmberPanId panId, const EmberEUI64 ieeeAddr, uint8_t nodeCapabilities, const EmberRf4ceVendorInfo *vendorInfo, const EmberRf4ceApplicationInfo *appInfo, uint8_t rxLinkQuality, uint8_t discRequestLqi)

- `bool emberAfPluginRf4ceProfileMsoDiscoveryResponseCallback (bool atCapacity, uint8_t channel, EmberPanId panId, const EmberEUI64 ieeeAddr, uint8_t nodeCapabilities, const EmberRf4ceVendorInfo *vendorInfo, const EmberRf4ceApplicationInfo *appInfo, uint8_t rxLinkQuality, uint8_t discRequestLqi)`
- `void emberAfPluginRf4ceProfileGdpDiscoveryCompleteCallback (EmberStatus status)`
- `void emberAfPluginRf4ceProfileRemoteControl11DiscoveryCompleteCallback (EmberStatus status)`
- `void emberAfPluginRf4ceProfileZrc20DiscoveryCompleteCallback (EmberStatus status)`
- `void emberAfPluginRf4ceProfileMsoDiscoveryCompleteCallback (EmberStatus status)`
- `void emberAfPluginRf4ceProfileGdpAutoDiscoveryResponseCompleteCallback (EmberStatus status, const EmberEUI64 srcIeeeAddr, uint8_t nodeCapabilities, const EmberRf4ceVendorInfo *vendorInfo, const EmberRf4ceApplicationInfo *appInfo, uint8_t searchDevType)`
- `void emberAfPluginRf4ceProfileRemoteControl11AutoDiscoveryResponseCompleteCallback (EmberStatus status, const EmberEUI64 srcIeeeAddr, uint8_t nodeCapabilities, const EmberRf4ceVendorInfo *vendorInfo, const EmberRf4ceApplicationInfo *appInfo, uint8_t searchDevType)`
- `void emberAfPluginRf4ceProfileZrc20AutoDiscoveryResponseCompleteCallback (EmberStatus status, const EmberEUI64 srcIeeeAddr, uint8_t nodeCapabilities, const EmberRf4ceVendorInfo *vendorInfo, const EmberRf4ceApplicationInfo *appInfo, uint8_t searchDevType)`
- `void emberAfPluginRf4ceProfileMsoAutoDiscoveryResponseCompleteCallback (EmberStatus status, const EmberEUI64 srcIeeeAddr, uint8_t nodeCapabilities, const EmberRf4ceVendorInfo *vendorInfo, const EmberRf4ceApplicationInfo *appInfo, uint8_t searchDevType)`
- `bool emberAfPluginRf4ceProfileGdpPairRequestCallback (EmberStatus status, uint8_t pairingIndex, const EmberEUI64 sourceIeeeAddr, uint8_t nodeCapabilities, const EmberRf4ceVendorInfo *vendorInfo, const EmberRf4ceApplicationInfo *appInfo, uint8_t keyExchangeTransferCount)`
- `bool emberAfPluginRf4ceProfileRemoteControl11PairRequestCallback (EmberStatus status, uint8_t pairingIndex, const EmberEUI64 sourceIeeeAddr, uint8_t nodeCapabilities, const EmberRf4ceVendorInfo *vendorInfo, const EmberRf4ceApplicationInfo *appInfo, uint8_t keyExchangeTransferCount)`
- `bool emberAfPluginRf4ceProfileZrc20PairRequestCallback (EmberStatus status, uint8_t pairingIndex, const EmberEUI64 sourceIeeeAddr, uint8_t nodeCapabilities, const EmberRf4ceVendorInfo *vendorInfo, const EmberRf4ceApplicationInfo *appInfo, uint8_t keyExchangeTransferCount)`
- `bool emberAfPluginRf4ceProfileMsoPairRequestCallback (EmberStatus status, uint8_t pairingIndex, const EmberEUI64 sourceIeeeAddr, uint8_t nodeCapabilities, const EmberRf4ceVendorInfo *vendorInfo, const EmberRf4ceApplicationInfo *appInfo, uint8_t keyExchangeTransferCount)`
- `void emberAfPluginRf4ceProfileGdpPairCompleteCallback (EmberStatus status, uint8_t pairingIndex, const EmberRf4ceVendorInfo *vendorInfo, const EmberRf4ceApplicationInfo *appInfo)`
- `void emberAfPluginRf4ceProfileRemoteControl11PairCompleteCallback (EmberStatus status, uint8_t pairingIndex, const EmberRf4ceVendorInfo *vendorInfo, const EmberRf4ceApplicationInfo *appInfo)`
- `void emberAfPluginRf4ceProfileZrc20PairCompleteCallback (EmberStatus status, uint8_t pairingIndex, const EmberRf4ceVendorInfo *vendorInfo, const EmberRf4ceApplicationInfo *appInfo)`
- `void emberAfPluginRf4ceProfileMsoPairCompleteCallback (EmberStatus status, uint8_t pairingIndex, const EmberRf4ceVendorInfo *vendorInfo, const EmberRf4ceApplicationInfo *appInfo)`

6.29.1 Detailed Description

These callbacks are contributed by the RF4CE Profile Support plugin.

6.29.2 Function Documentation

6.29.2.1 bool emberAfPluginRf4ceProfileGdpMessageSentCallback (uint8_t pairingIndex, uint8_t profileId, uint16_t vendorId, uint8_t messageTag, const uint8_t * message, uint8_t messageLength, EmberStatus status)

Gdp Message Sent.

This function is called by the RF4CE Profile plugin when a GDP message is sent. It is expected to return true if the sent message was processed by the GDP plugin, false otherwise.

Parameters

<i>pairingIndex</i>	The index of the entry in the pairing table used to transmit the message. Ver.: always
<i>profileId</i>	The profile ID included in the message. Ver.: always
<i>vendorId</i>	The vendor ID included in the message. Ver.: always
<i>messageTag</i>	The tag of the message. Ver.: always
<i>message</i>	A pointer to the payload of the message that was sent. Ver.: always
<i>messageLength</i>	The length in bytes of the message. Ver.: always
<i>status</i>	An ::EmberStatus value. Ver.: always

Definition at line 1529 of file [callback.doc](#).

6.29.2.2 void emberAfPluginRf4ceProfileRemoteControl11MessageSentCallback (uint8_t pairingIndex, uint16_t vendorId, uint8_t messageTag, const uint8_t * message, uint8_t messageLength, EmberStatus status)

Remote Control 1 1 Message Sent.

This function is called by the RF4CE Profile plugin when a Remote Control 1.0 or 1.1 message is sent.

Parameters

<i>pairingIndex</i>	The index of the entry in the pairing table used to transmit the message. Ver.: always
<i>vendorId</i>	The vendor ID included in the message. Ver.: always
<i>messageTag</i>	The tag of the message. Ver.: always
<i>message</i>	A pointer to the payload of the message that was sent. Ver.: always
<i>messageLength</i>	The length in bytes of the message. Ver.: always
<i>status</i>	An ::EmberStatus value. Ver.: always

Definition at line 1554 of file [callback.doc](#).

6.29.2.3 void emberAfPluginRf4ceProfileZrc20MessageSentCallback (uint8_t pairingIndex, uint16_t vendorId, uint8_t messageTag, const uint8_t * message, uint8_t messageLength, EmberStatus status)

Zrc 2 0 Message Sent.

This function is called by the RF4CE Profile plugin when a ZRC 2.0 message is sent.

Parameters

<i>pairingIndex</i>	The index of the entry in the pairing table used to transmit the message. Ver.: always
<i>vendorId</i>	The vendor ID included in the message. Ver.: always
<i>messageTag</i>	The tag of the message. Ver.: always
<i>message</i>	A pointer to the payload of the message that was sent. Ver.: always

<i>messageLength</i>	The length in bytes of the message. Ver.: always
<i>status</i>	An ::EmberStatus value. Ver.: always

Definition at line 1577 of file [callback.doc](#).

6.29.2.4 void emberAfPluginRf4ceProfileMsoMessageSentCallback (uint8_t *pairingIndex*, uint16_t *vendorId*, uint8_t *messageTag*, const uint8_t * *message*, uint8_t *messageLength*, EmberStatus *status*)

Mso Message Sent.

This function is called by the RF4CE Profile plugin when an MSO message is sent.

Parameters

<i>pairingIndex</i>	The index of the entry in the pairing table used to transmit the message. Ver.: always
<i>vendorId</i>	The vendor ID included in the message. Ver.: always
<i>messageTag</i>	The tag of the message. Ver.: always
<i>message</i>	A pointer to the payload of the message that was sent. Ver.: always
<i>messageLength</i>	The length in bytes of the message. Ver.: always
<i>status</i>	An ::EmberStatus value. Ver.: always

Definition at line 1600 of file [callback.doc](#).

6.29.2.5 void emberAfPluginRf4ceProfileMessageSentCallback (uint8_t *pairingIndex*, uint8_t *profileId*, uint16_t *vendorId*, uint8_t *messageTag*, const uint8_t * *message*, uint8_t *messageLength*, EmberStatus *status*)

Message Sent.

This function is called by the RF4CE Profile plugin when a message is sent. It is called after any profile-specific callbacks.

Parameters

<i>pairingIndex</i>	The index of the entry in the pairing table used to transmit the message. Ver.: always
<i>profileId</i>	The profile ID included in the message. Ver.: always
<i>vendorId</i>	The vendor ID included in the message. Ver.: always
<i>messageTag</i>	The tag of the message. Ver.: always
<i>message</i>	A pointer to the payload of the message that was sent. Ver.: always
<i>messageLength</i>	The length in bytes of the message. Ver.: always
<i>status</i>	An ::EmberStatus value. Ver.: always

Definition at line 1624 of file [callback.doc](#).

6.29.2.6 bool emberAfPluginRf4ceProfileGdpIncomingMessageCallback (uint8_t *pairingIndex*, uint8_t *profileId*, uint16_t *vendorId*, EmberRf4ceTxOption *txOptions*, const uint8_t * *message*, uint8_t *messageLength*)

Gdp Incoming Message.

This function is called by the RF4CE Profile plugin when any message is received. This callback is expected to return true if the passed message was processed by the GDP plugin, false otherwise.

Parameters

<i>pairingIndex</i>	The index of the entry in the pairing table corresponding to the PAN on which the message was received. Ver.: always
<i>profileId</i>	The profile ID included in the message. Ver.: always
<i>vendorId</i>	The vendor ID included in the message. Ver.: always
<i>txOptions</i>	The TX options used by the source node to transmit the received message. Ver.: always
<i>message</i>	A pointer to the payload of the received message. Ver.: always
<i>messageLength</i>	The length in bytes of the received message. Ver.: always

Definition at line 1651 of file [callback.doc](#).

6.29.2.7 void emberAfPluginRf4ceProfileRemoteControl11IncomingMessageCallback (uint8_t pairingIndex, uint16_t vendorId, EmberRf4ceTxOption txOptions, const uint8_t * message, uint8_t messageLength)

Remote Control 1 1 Incoming Message.

This function is called by the RF4CE Profile plugin when a Remote Control 1.0 or 1.1 message is received.

Parameters

<i>pairingIndex</i>	The index of the entry in the pairing table corresponding to the PAN on which the message was received. Ver.: always
<i>vendorId</i>	The vendor ID included in the message. Ver.: always
<i>txOptions</i>	The TX options used by the source node to transmit the received message. Ver.: always
<i>message</i>	A pointer to the payload of the received message. Ver.: always
<i>messageLength</i>	The length in bytes of the received message. Ver.: always

Definition at line 1676 of file [callback.doc](#).

6.29.2.8 void emberAfPluginRf4ceProfileZrc20IncomingMessageCallback (uint8_t pairingIndex, uint16_t vendorId, EmberRf4ceTxOption txOptions, const uint8_t * message, uint8_t messageLength)

Zrc 2 0 Incoming Message.

This function is called by the RF4CE Profile plugin when a ZRC 2.0 message is received.

Parameters

<i>pairingIndex</i>	The index of the entry in the pairing table corresponding to the PAN on which the message was received. Ver.: always
<i>vendorId</i>	The vendor ID included in the message. Ver.: always
<i>txOptions</i>	The TX options used by the source node to transmit the received message. Ver.: always
<i>message</i>	A pointer to the payload of the received message. Ver.: always
<i>messageLength</i>	The length in bytes of the received message. Ver.: always

Definition at line 1699 of file [callback.doc](#).

6.29.2.9 void emberAfPluginRf4ceProfileMsoIncomingMessageCallback (*uint8_t pairingIndex, uint16_t vendorId, EmberRf4ceTxOption txOptions, const uint8_t * message, uint8_t messageLength*)

Mso Incoming Message.

This function is called by the RF4CE Profile plugin when an MSO message is received.

Parameters

<i>pairingIndex</i>	The index of the entry in the pairing table corresponding to the PAN on which the message was received. Ver.: always
<i>vendorId</i>	The vendor ID included in the message. Ver.: always
<i>txOptions</i>	The TX options used by the source node to transmit the received message. Ver.: always
<i>message</i>	A pointer to the payload of the received message. Ver.: always
<i>messageLength</i>	The length in bytes of the received message. Ver.: always

Definition at line 1722 of file [callback.doc](#).

6.29.2.10 void emberAfPluginRf4ceProfileIncomingMessageCallback (*uint8_t pairingIndex, uint8_t profileId, uint16_t vendorId, EmberRf4ceTxOption txOptions, const uint8_t * message, uint8_t messageLength*)

Incoming Message.

This function is called by the RF4CE Profile plugin when a message is received. It is called after any profile-specific callbacks.

Parameters

<i>pairingIndex</i>	The index of the entry in the pairing table corresponding to the PAN on which the message was received. Ver.: always
<i>profileId</i>	The profile ID included in the message. Ver.: always
<i>vendorId</i>	The vendor ID included in the message. Ver.: always
<i>txOptions</i>	The TX options used by the source node to transmit the received message. Ver.: always
<i>message</i>	A pointer to the payload of the received message. Ver.: always
<i>messageLength</i>	The length in bytes of the received message. Ver.: always

Definition at line 1746 of file [callback.doc](#).

6.29.2.11 bool emberAfPluginRf4ceProfileGdpDiscoveryRequestCallback (*const EmberEUI64 ieeeAddr, uint8_t nodeCapabilities, const EmberRf4ceVendorInfo * vendorInfo, const EmberRf4ceApplicationInfo * applInfo, uint8_t searchDevType, uint8_t rxLinkQuality*)

Gdp Discovery Request.

This function is called by the RF4CE Profile plugin when a discovery request for the GDP profile is received.

Parameters

<i>ieeeAddr</i>	The IEEE address of the requesting device. Ver.: always
<i>node-Capabilities</i>	The capabilities of the requesting device. Ver.: always
<i>vendorInfo</i>	The vendor information of the requesting device. Ver.: always

<i>appInfo</i>	The application information of the requesting device. Ver.: always
<i>searchDevType</i>	The device type requested. Ver.: always
<i>rxLinkQuality</i>	The LQI of the request. Ver.: always

Definition at line 1770 of file [callback.doc](#).

6.29.2.12 bool emberAfPluginRf4ceProfileRemoteControl11DiscoveryRequestCallback (const EmberEUI64 *ieeeAddr*, uint8_t *nodeCapabilities*, const EmberRf4ceVendorInfo * *vendorInfo*, const EmberRf4ceApplicationInfo * *appInfo*, uint8_t *searchDevType*, uint8_t *rxLinkQuality*)

Remote Control 1 1 Discovery Request.

This function is called by the RF4CE Profile plugin when a discovery request for the Remote Control 1.0 or 1.1 profile is received.

Parameters

<i>ieeeAddr</i>	The IEEE address of the requesting device. Ver.: always
<i>node- Capabilities</i>	The capabilities of the requesting device. Ver.: always
<i>vendorInfo</i>	The vendor information of the requesting device. Ver.: always
<i>appInfo</i>	The application information of the requesting device. Ver.: always
<i>searchDevType</i>	The device type requested. Ver.: always
<i>rxLinkQuality</i>	The LQI of the request. Ver.: always

Definition at line 1795 of file [callback.doc](#).

6.29.2.13 bool emberAfPluginRf4ceProfileZrc20DiscoveryRequestCallback (const EmberEUI64 *ieeeAddr*, uint8_t *nodeCapabilities*, const EmberRf4ceVendorInfo * *vendorInfo*, const EmberRf4ceApplicationInfo * *appInfo*, uint8_t *searchDevType*, uint8_t *rxLinkQuality*)

Zrc 2 0 Discovery Request.

This function is called by the RF4CE Profile plugin when a discovery request for the ZRC 2.0 profile is received.

Parameters

<i>ieeeAddr</i>	The IEEE address of the requesting device. Ver.: always
<i>node- Capabilities</i>	The capabilities of the requesting device. Ver.: always
<i>vendorInfo</i>	The vendor information of the requesting device. Ver.: always
<i>appInfo</i>	The application information of the requesting device. Ver.: always
<i>searchDevType</i>	The device type requested. Ver.: always
<i>rxLinkQuality</i>	The LQI of the request. Ver.: always

Definition at line 1820 of file [callback.doc](#).

6.29.2.14 bool emberAfPluginRf4ceProfileMsoDiscoveryRequestCallback (const EmberEUI64 *ieeeAddr*, uint8_t *nodeCapabilities*, const EmberRf4ceVendorInfo * *vendorInfo*, const EmberRf4ceApplicationInfo * *appInfo*, uint8_t *searchDevType*, uint8_t *rxLinkQuality*)

Mso Discovery Request.

This function is called by the RF4CE Profile plugin when a discovery request for the MSO profile is received.

Parameters

<i>ieeeAddr</i>	The IEEE address of the requesting device. Ver.: always
<i>nodeCapabilities</i>	The capabilities of the requesting device. Ver.: always
<i>vendorInfo</i>	The vendor information of the requesting device. Ver.: always
<i>appInfo</i>	The application information of the requesting device. Ver.: always
<i>searchDevType</i>	The device type requested. Ver.: always
<i>rxLinkQuality</i>	The LQI of the request. Ver.: always

Definition at line 1845 of file [callback.doc](#).

6.29.2.15 bool emberAfPluginRf4ceProfileGdpDiscoveryResponseCallback (bool *atCapacity*, uint8_t *channel*, EmberPanId *panId*, const EmberEUI64 *ieeeAddr*, uint8_t *nodeCapabilities*, const EmberRf4ceVendorInfo * *vendorInfo*, const EmberRf4ceApplicationInfo * *appInfo*, uint8_t *rxLinkQuality*, uint8_t *discRequestLqi*)

Gdp Discovery Response.

This function is called by the RF4CE Profile plugin when a discovery response for the GDP profile is received.

Parameters

<i>atCapacity</i>	true if the node sending the discovery response has no free entry in its pairing table, false otherwise. Ver.: always
<i>channel</i>	The channel on which the discovery response was received. Ver.: always
<i>panId</i>	The PAN identifier of the responding device. Ver.: always
<i>ieeeAddr</i>	The IEEE address of the responding device. Ver.: always
<i>nodeCapabilities</i>	The capabilities of the responding device. Ver.: always
<i>vendorInfo</i>	The vendor information of the responding device. Ver.: always
<i>appInfo</i>	The application information of the responding device. Ver.: always
<i>rxLinkQuality</i>	The LQI of the discovery response. Ver.: always
<i>discRequestLqi</i>	The LQI of the discovery request command frame reported by the responding device. Ver.: always

Definition at line 1876 of file [callback.doc](#).

6.29.2.16 bool emberAfPluginRf4ceProfileRemoteControl11DiscoveryResponseCallback (bool *atCapacity*, uint8_t *channel*, EmberPanId *panId*, const EmberEUI64 *ieeeAddr*, uint8_t *nodeCapabilities*, const EmberRf4ceVendorInfo * *vendorInfo*, const EmberRf4ceApplicationInfo * *appInfo*, uint8_t *rxLinkQuality*, uint8_t *discRequestLqi*)

Remote Control 1 1 Discovery Response.

This function is called by the RF4CE Profile plugin when a discovery response for the Remote Control 1.0 or 1.1 profile is received.

Parameters

<i>atCapacity</i>	true if the node sending the discovery response has no free entry in its pairing table, false otherwise. Ver.: always
<i>channel</i>	The channel on which the discovery response was received. Ver.: always
<i>panId</i>	The PAN identifier of the responding device. Ver.: always
<i>ieeeAddr</i>	The IEEE address of the responding device. Ver.: always
<i>node-Capabilities</i>	The capabilities of the responding device. Ver.: always
<i>vendorInfo</i>	The vendor information of the responding device. Ver.: always
<i>appInfo</i>	The application information of the responding device. Ver.: always
<i>rxLinkQuality</i>	The LQI of the discovery response. Ver.: always
<i>discRequestLqi</i>	The LQI of the discovery request command frame reported by the responding device. Ver.: always

Definition at line 1910 of file [callback.doc](#).

6.29.2.17 `bool emberAfPluginRf4ceProfileZrc20DiscoveryResponseCallback (bool atCapacity, uint8_t channel, EmberPanId panId, const EmberEUI64 ieeeAddr, uint8_t nodeCapabilities, const EmberRf4ceVendorInfo * vendorInfo, const EmberRf4ceApplicationInfo * appInfo, uint8_t rxLinkQuality, uint8_t discRequestLqi)`

Zrc 2 0 Discovery Response.

This function is called by the RF4CE Profile plugin when a discovery response for the ZRC 2.0 profile is received.

Parameters

<i>atCapacity</i>	true if the node sending the discovery response has no free entry in its pairing table, false otherwise. Ver.: always
<i>channel</i>	The channel on which the discovery response was received. Ver.: always
<i>panId</i>	The PAN identifier of the responding device. Ver.: always
<i>ieeeAddr</i>	The IEEE address of the responding device. Ver.: always
<i>node-Capabilities</i>	The capabilities of the responding device. Ver.: always
<i>vendorInfo</i>	The vendor information of the responding device. Ver.: always
<i>appInfo</i>	The application information of the responding device. Ver.: always
<i>rxLinkQuality</i>	The LQI of the discovery response. Ver.: always
<i>discRequestLqi</i>	The LQI of the discovery request command frame reported by the responding device. Ver.: always

Definition at line 1944 of file [callback.doc](#).

6.29.2.18 `bool emberAfPluginRf4ceProfileMsoDiscoveryResponseCallback (bool atCapacity, uint8_t channel, EmberPanId panId, const EmberEUI64 ieeeAddr, uint8_t nodeCapabilities, const EmberRf4ceVendorInfo * vendorInfo, const EmberRf4ceApplicationInfo * appInfo, uint8_t rxLinkQuality, uint8_t discRequestLqi)`

Mso Discovery Response.

This function is called by the RF4CE Profile plugin when a discovery response for the MSO profile is received.

Parameters

<i>atCapacity</i>	true if the node sending the discovery response has no free entry in its pairing table, false otherwise. Ver.: always
<i>channel</i>	The channel on which the discovery response was received. Ver.: always
<i>panId</i>	The PAN identifier of the responding device. Ver.: always
<i>ieeeAddr</i>	The IEEE address of the responding device. Ver.: always
<i>node-Capabilities</i>	The capabilities of the responding device. Ver.: always
<i>vendorInfo</i>	The vendor information of the responding device. Ver.: always
<i>appInfo</i>	The application information of the responding device. Ver.: always
<i>rxLinkQuality</i>	The LQI of the discovery response. Ver.: always
<i>discRequestLqi</i>	The LQI of the discovery request command frame reported by the responding device. Ver.: always

Definition at line 1978 of file [callback.doc](#).

6.29.2.19 void emberAfPluginRf4ceProfileGdpDiscoveryCompleteCallback (EmberStatus *status*)

Gdp Discovery Complete.

This function is called by the RF4CE Profile plugin when the discovery process for the GDP profile has completed.

Parameters

<i>status</i>	An ::EmberStatus value indicating whether the discovery succeeded or the failure reason. Ver.: always
---------------	---

Definition at line 1999 of file [callback.doc](#).

6.29.2.20 void emberAfPluginRf4ceProfileRemoteControl11DiscoveryCompleteCallback (EmberStatus *status*)

Remote Control 1 1 Discovery Complete.

This function is called by the RF4CE Profile plugin when the discovery process for the Remote Control 1.0 or 1.1 profile has completed.

Parameters

<i>status</i>	An ::EmberStatus value indicating whether the discovery succeeded or the failure reason. Ver.: always
---------------	---

Definition at line 2011 of file [callback.doc](#).

6.29.2.21 void emberAfPluginRf4ceProfileZrc20DiscoveryCompleteCallback (EmberStatus *status*)

Zrc 2 0 Discovery Complete.

This function is called by the RF4CE Profile plugin when the discovery process for the ZRC 2.0 profile has completed.

Parameters

<i>status</i>	An ::EmberStatus value indicating whether the discovery succeeded or the failure reason. Ver.: always
---------------	---

Definition at line 2023 of file [callback.doc](#).

6.29.2.22 void emberAfPluginRf4ceProfileMsoDiscoveryCompleteCallback (EmberStatus *status*)

Mso Discovery Complete.

This function is called by the RF4CE Profile plugin when the discovery process for the MSO profile has completed.

Parameters

<i>status</i>	An ::EmberStatus value indicating whether the discovery succeeded or the failure reason. Ver.: always
---------------	---

Definition at line 2035 of file [callback.doc](#).

6.29.2.23 void emberAfPluginRf4ceProfileGdpAutoDiscoveryResponseCompleteCallback (EmberStatus *status*, const EmberEUI64 *srcIEEEAddr*, uint8_t *nodeCapabilities*, const EmberRf4ceVendorInfo * *vendorInfo*, const EmberRf4ceApplicationInfo * *appInfo*, uint8_t *searchDevType*)

Gdp Auto Discovery Response Complete.

This function is called by the RF4CE Profile plugin when the auto discovery response process for the GDP profile has completed.

Parameters

<i>status</i>	An ::EmberStatus value indicating whether the discovery succeeded or the failure reason. Ver.: always
<i>srcIEEEAddr</i>	The IEEE address of the node from which the discovery request command frame was received. This parameter is meaningful only if the status parameter is ::EMBER_SUCCESS. Ver.: always
<i>nodeCapabilities</i>	The node capabilities of the node that issued the discovery request. This parameter is meaningful only if the status parameter is ::EMBER_SUCCESS. Ver.: always
<i>vendorInfo</i>	A pointer to an ::EmberRf4ceVendorInfo struct containing the vendor information of the node that issued the discovery request. This parameter is non-NULL only if the status parameter is ::EMBER_SUCCESS. Ver.: always
<i>appInfo</i>	A pointer to an ::EmberRf4ceApplicationInfo struct containing the application information of the node that issued the discovery request. This parameter is non-NULL only if the status parameter is ::EMBER_SUCCESS. Ver.: always
<i>searchDevType</i>	The device type being discovered. If this is 0xFF, any type is being requested. This parameter is meaningful only if the status parameter is ::EMBER_SUCCESS. Ver.: always

Definition at line 2064 of file [callback.doc](#).

6.29.2.24 void emberAfPluginRf4ceProfileRemoteControl11AutoDiscoveryResponseCompleteCallback (EmberStatus status, const EmberEUI64 srcIeeeAddr, uint8_t nodeCapabilities, const EmberRf4ceVendorInfo * vendorInfo, const EmberRf4ceApplicationInfo * appInfo, uint8_t searchDevType)

Remote Control 1 1 Auto Discovery Response Complete.

This function is called by the RF4CE Profile plugin when the auto discovery response process for the Remote Control 1.0 or 1.1 profile has completed.

Parameters

<i>status</i>	An ::EmberStatus value indicating whether the discovery succeeded or the failure reason. Ver.: always
<i>srcIeeeAddr</i>	The IEEE address of the node from which the discovery request command frame was received. This parameter is meaningful only if the status parameter is ::EMBER_SUCCESS. Ver.: always
<i>nodeCapabilities</i>	The node capabilities of the node that issued the discovery request. This parameter is meaningful only if the status parameter is ::EMBER_SUCCESS. Ver.: always
<i>vendorInfo</i>	A pointer to an ::EmberRf4ceVendorInfo struct containing the vendor information of the node that issued the discovery request. This parameter is non-NULL only if the status parameter is ::EMBER_SUCCESS. Ver.: always
<i>appInfo</i>	A pointer to an ::EmberRf4ceApplicationInfo struct containing the application information of the node that issued the discovery request. This parameter is non-NULL only if the status parameter is ::EMBER_SUCCESS. Ver.: always
<i>searchDevType</i>	The device type being discovered. If this is 0xFF, any type is being requested. This parameter is meaningful only if the status parameter is ::EMBER_SUCCESS. Ver.: always

Definition at line 2098 of file [callback.doc](#).

6.29.2.25 void emberAfPluginRf4ceProfileZrc20AutoDiscoveryResponseCompleteCallback (EmberStatus status, const EmberEUI64 srcIeeeAddr, uint8_t nodeCapabilities, const EmberRf4ceVendorInfo * vendorInfo, const EmberRf4ceApplicationInfo * appInfo, uint8_t searchDevType)

Zrc 2 0 Auto Discovery Response Complete.

This function is called by the RF4CE Profile plugin when the auto discovery response process for the ZRC 2.0 profile has completed.

Parameters

<i>status</i>	An ::EmberStatus value indicating whether the discovery succeeded or the failure reason. Ver.: always
<i>srcIeeeAddr</i>	The IEEE address of the node from which the discovery request command frame was received. This parameter is meaningful only if the status parameter is ::EMBER_SUCCESS. Ver.: always
<i>nodeCapabilities</i>	The node capabilities of the node that issued the discovery request. This parameter is meaningful only if the status parameter is ::EMBER_SUCCESS. Ver.: always
<i>vendorInfo</i>	A pointer to an ::EmberRf4ceVendorInfo struct containing the vendor information of the node that issued the discovery request. This parameter is non-NULL only if the status parameter is ::EMBER_SUCCESS. Ver.: always
<i>appInfo</i>	A pointer to an ::EmberRf4ceApplicationInfo struct containing the application information of the node that issued the discovery request. This parameter is non-NULL only if the status parameter is ::EMBER_SUCCESS. Ver.: always

<i>searchDevType</i>	The device type being discovered. If this is 0xFF, any type is being requested. This parameter is meaningful only if the status parameter is ::EMBER_SUCCESS. Ver.: always
----------------------	--

Definition at line 2132 of file [callback.doc](#).

6.29.2.26 void emberAfPluginRf4ceProfileMsoAutoDiscoveryResponseCompleteCallback (EmberStatus status, const EmberEUI64 srcIEEEAddr, uint8_t nodeCapabilities, const EmberRf4ceVendorInfo * vendorInfo, const EmberRf4ceApplicationInfo * appInfo, uint8_t searchDevType)

Mso Auto Discovery Response Complete.

This function is called by the RF4CE Profile plugin when the auto discovery response process for the MSO profile has completed.

Parameters

<i>status</i>	An ::EmberStatus value indicating whether the discovery succeeded or the failure reason. Ver.: always
<i>srcIEEEAddr</i>	The IEEE address of the node from which the discovery request command frame was received. This parameter is meaningful only if the status parameter is ::EMBER_SUCCESS Ver.: always
<i>nodeCapabilities</i>	The node capabilities of the node that issued the discovery request. This parameter is meaningful only if the status parameter is ::EMBER_SUCCESS. Ver.: always
<i>vendorInfo</i>	A pointer to an ::EmberRf4ceVendorInfo struct containing the vendor information of the node that issued the discovery request. This parameter is non-NULL only if the status parameter is ::EMBER_SUCCESS. Ver.: always
<i>appInfo</i>	A pointer to an ::EmberRf4ceApplicationInfo struct containing the application information of the node that issued the discovery request. This parameter is non-NULL only if the status parameter is ::EMBER_SUCCESS. Ver.: always
<i>searchDevType</i>	The device type being discovered. If this is 0xFF, any type is being requested. This parameter is meaningful only if the status parameter is ::EMBER_SUCCESS. Ver.: always

Definition at line 2166 of file [callback.doc](#).

6.29.2.27 bool emberAfPluginRf4ceProfileGdpPairRequestCallback (EmberStatus status, uint8_t pairingIndex, const EmberEUI64 sourceIEEEAddr, uint8_t nodeCapabilities, const EmberRf4ceVendorInfo * vendorInfo, const EmberRf4ceApplicationInfo * appInfo, uint8_t keyExchangeTransferCount)

Gdp Pair Request.

This function is called by the RF4CE Profile plugin when a pair request for the GDP profile is received.

Parameters

<i>status</i>	An ::EmberStatus value indicating whether the pairing is a new pairing, a duplicate or if the pairing table is full. Ver.: always
<i>pairingIndex</i>	The index of the entry that will be used by the stack for the pairing link. Ver.: always
<i>sourceIEEEAddr</i>	The source IEEE address of the incoming pair request. Ver.: always
<i>nodeCapabilities</i>	The node capabilities of the node that issued the pair request. Ver.: always

<i>vendorInfo</i>	The vendor information of the requesting device. Ver.: always
<i>appInfo</i>	The application information of the requesting device. Ver.: always
<i>keyExchangeTransferCount</i>	The number of transfers to be used to exchange the link key with the pairing originator. Ver.: always

Definition at line 2195 of file [callback.doc](#).

6.29.2.28 `bool emberAfPluginRf4ceProfileRemoteControl11PairRequestCallback (EmberStatus status, uint8_t pairingIndex, const EmberEUI64 sourceIeeeAddr, uint8_t nodeCapabilities, const EmberRf4ceVendorInfo * vendorInfo, const EmberRf4ceApplicationInfo * appInfo, uint8_t keyExchangeTransferCount)`

Remote Control 1 1 Pair Request.

This function is called by the RF4CE Profile plugin when a pair request for the Remote Control 1.0 or 1.1 profile is received.

Parameters

<i>status</i>	An ::EmberStatus value indicating whether the pairing is a new pairing, a duplicate or if the pairing table is full. Ver.: always
<i>pairingIndex</i>	The index of the entry that will be used by the stack for the pairing link. Ver.: always
<i>sourceIeeeAddr</i>	The source IEEE address of the incoming pair request. Ver.: always
<i>nodeCapabilities</i>	The node capabilities of the node that issued the pair request. Ver.: always
<i>vendorInfo</i>	The vendor information of the requesting device. Ver.: always
<i>appInfo</i>	The application information of the requesting device. Ver.: always
<i>keyExchangeTransferCount</i>	The number of transfers to be used to exchange the link key with the pairing originator. Ver.: always

Definition at line 2226 of file [callback.doc](#).

6.29.2.29 `bool emberAfPluginRf4ceProfileZrc20PairRequestCallback (EmberStatus status, uint8_t pairingIndex, const EmberEUI64 sourceIeeeAddr, uint8_t nodeCapabilities, const EmberRf4ceVendorInfo * vendorInfo, const EmberRf4ceApplicationInfo * appInfo, uint8_t keyExchangeTransferCount)`

Zrc 2 0 Pair Request.

This function is called by the RF4CE Profile plugin when a pair request for the ZRC 2.0 profile is received.

Parameters

<i>status</i>	An ::EmberStatus value indicating whether the pairing is a new pairing, a duplicate or if the pairing table is full. Ver.: always
<i>pairingIndex</i>	The index of the entry that will be used by the stack for the pairing link. Ver.: always
<i>sourceIeeeAddr</i>	The source IEEE address of the incoming pair request. Ver.: always
<i>nodeCapabilities</i>	The node capabilities of the node that issued the pair request. Ver.: always
<i>vendorInfo</i>	The vendor information of the requesting device. Ver.: always
<i>appInfo</i>	The application information of the requesting device. Ver.: always
<i>keyExchangeTransferCount</i>	The number of transfers to be used to exchange the link key with the pairing originator. Ver.: always

Definition at line 2257 of file [callback.doc](#).

6.29.2.30 bool emberAfPluginRf4ceProfileMsoPairRequestCallback (EmberStatus status, uint8_t pairingIndex, const EmberEUI64 sourceIeeeAddr, uint8_t nodeCapabilities, const EmberRf4ceVendorInfo * vendorInfo, const EmberRf4ceApplicationInfo * appInfo, uint8_t keyExchangeTransferCount)

Mso Pair Request.

This function is called by the RF4CE Profile plugin when a pair request for the MSO profile is received.

Parameters

<i>status</i>	An ::EmberStatus value indicating whether the pairing is a new pairing, a duplicate or if the pairing table is full. Ver.: always
<i>pairingIndex</i>	The index of the entry that will be used by the stack for the pairing link. Ver.: always
<i>sourceIeeeAddr</i>	The source IEEE address of the incoming pair request. Ver.: always
<i>nodeCapabilities</i>	The node capabilities of the node that issued the pair request. Ver.: always
<i>vendorInfo</i>	The vendor information of the requesting device. Ver.: always
<i>appInfo</i>	The application information of the requesting device. Ver.: always
<i>keyExchangeTransferCount</i>	The number of transfers to be used to exchange the link key with the pairing originator. Ver.: always

Definition at line 2288 of file [callback.doc](#).

6.29.2.31 void emberAfPluginRf4ceProfileGdpPairCompleteCallback (EmberStatus status, uint8_t pairingIndex, const EmberRf4ceVendorInfo * vendorInfo, const EmberRf4ceApplicationInfo * appInfo)

Gdp Pair Complete.

This function is called by the RF4CE Profile plugin when the pairing process for the GDP profile has completed.

Parameters

<i>status</i>	An ::EmberStatus value indicating whether the pairing succeeded or the failure reason. Ver.: always
<i>pairingIndex</i>	The index of the pairing table entry corresponding to the pairing link that was established during the pairing process. Ver.: always
<i>vendorInfo</i>	The vendor information of the peer device. Ver.: always
<i>appInfo</i>	The application information of the peer device. Ver.: always

Definition at line 2311 of file [callback.doc](#).

6.29.2.32 void emberAfPluginRf4ceProfileRemoteControl11PairCompleteCallback (EmberStatus status, uint8_t pairingIndex, const EmberRf4ceVendorInfo * vendorInfo, const EmberRf4ceApplicationInfo * appInfo)

Remote Control 1 1 Pair Complete.

This function is called by the RF4CE Profile plugin when the pairing process for the Remote Control 1.0 or 1.1 profile has completed.

Parameters

<i>status</i>	An ::EmberStatus value indicating whether the pairing succeeded or the failure reason. Ver.: always
<i>pairingIndex</i>	The index of the pairing table entry corresponding to the pairing link that was established during the pairing process. Ver.: always
<i>vendorInfo</i>	The vendor information of the peer device. Ver.: always
<i>appInfo</i>	The application information of the peer device. Ver.: always

Definition at line 2330 of file [callback.doc](#).

6.29.2.33 void emberAfPluginRf4ceProfileZrc20PairCompleteCallback (EmberStatus status, uint8_t pairingIndex, const EmberRf4ceVendorInfo * vendorInfo, const EmberRf4ceApplicationInfo * appInfo)

Zrc 2 0 Pair Complete.

This function is called by the RF4CE Profile plugin when the pairing process for the ZRC 2.0 profile has completed.

Parameters

<i>status</i>	An ::EmberStatus value indicating whether the pairing succeeded or the failure reason. Ver.: always
<i>pairingIndex</i>	The index of the pairing table entry corresponding to the pairing link that was established during the pairing process. Ver.: always
<i>vendorInfo</i>	The vendor information of the peer device. Ver.: always
<i>appInfo</i>	The application information of the peer device. Ver.: always

Definition at line 2349 of file [callback.doc](#).

6.29.2.34 void emberAfPluginRf4ceProfileMsoPairCompleteCallback (EmberStatus status, uint8_t pairingIndex, const EmberRf4ceVendorInfo * vendorInfo, const EmberRf4ceApplicationInfo * appInfo)

Mso Pair Complete.

This function is called by the RF4CE Profile plugin when the pairing process for the MSO profile has completed.

Parameters

<i>status</i>	An ::EmberStatus value indicating whether the pairing succeeded or the failure reason. Ver.: always
<i>pairingIndex</i>	The index of the pairing table entry corresponding to the pairing link that was established during the pairing process. Ver.: always
<i>vendorInfo</i>	The vendor information of the peer device. Ver.: always
<i>appInfo</i>	The application information of the peer device. Ver.: always

Definition at line 2368 of file [callback.doc](#).

6.30 RF4CE Stack Library Callbacks

Functions

- void `emberRf4ceMessageSentHandler` (EmberStatus status, uint8_t pairingIndex, uint8_t txOptions, uint8_t profileId, uint16_t vendorId, uint8_t messageTag, uint8_t messageLength, uint8_t *message)
- void `emberRf4ceIncomingMessageHandler` (uint8_t pairingIndex, uint8_t profileId, uint16_t vendorId, EmberRf4ceTxOption txOptions, uint8_t messageLength, uint8_t *message)
- void `emberRf4ceDiscoveryCompleteHandler` (EmberStatus status)
- bool `emberRf4ceDiscoveryRequestHandler` (EmberEUI64 srcIeeeAddr, uint8_t nodeCapabilities, EmberRf4ceVendorInfo *vendorInfo, EmberRf4ceApplicationInfo *appInfo, uint8_t searchDevType, uint8_t rxLinkQuality)
- bool `emberRf4ceDiscoveryResponseHandler` (bool atCapacity, uint8_t channel, EmberPanId panId, EmberEUI64 srcIeeeAddr, uint8_t nodeCapabilities, EmberRf4ceVendorInfo *vendorInfo, EmberRf4ceApplicationInfo *appInfo, uint8_t rxLinkQuality, uint8_t discRequestLqi)
- void `emberRf4ceAutoDiscoveryResponseCompleteHandler` (EmberStatus status, EmberEUI64 srcIeeeAddr, uint8_t nodeCapabilities, EmberRf4ceVendorInfo *vendorInfo, EmberRf4ceApplicationInfo *appInfo, uint8_t searchDevType)
- void `emberRf4cePairCompleteHandler` (EmberStatus status, uint8_t pairingIndex, EmberRf4ceVendorInfo *vendorInfo, EmberRf4ceApplicationInfo *appInfo)
- bool `emberRf4cePairRequestHandler` (EmberStatus status, uint8_t pairingIndex, EmberEUI64 srcIeeeAddr, uint8_t nodeCapabilities, EmberRf4ceVendorInfo *vendorInfo, EmberRf4ceApplicationInfo *appInfo, uint8_t keyExchangeTransferCount)
- void `emberRf4ceUnpairHandler` (uint8_t pairingIndex)
- void `emberRf4ceUnpairCompleteHandler` (uint8_t pairingIndex)

6.30.1 Detailed Description

These callbacks are contributed by the RF4CE Stack Library plugin.

6.30.2 Function Documentation

6.30.2.1 void `emberRf4ceMessageSentHandler` (EmberStatus *status*, uint8_t *pairingIndex*, uint8_t *txOptions*, uint8_t *profileId*, uint16_t *vendorId*, uint8_t *messageTag*, uint8_t *messageLength*, uint8_t * *message*)

A callback invoked by the ZigBee RF4CE stack when it has completed sending a message.

Parameters

<i>status</i>	An ::EmberStatus value of: <ul style="list-style-type: none"> • ::EMBER_SUCCESS - The message was successfully delivered. • ::EMBER_DELIVERY_FAILED - The message was not delivered.
<i>pairingIndex</i>	The index of the entry in the pairing table used to transmit the message.
<i>txOptions</i>	The TX options bitmask as per ZigBee RF4CE specification used for transmitting the packet.
<i>profileId</i>	The profile ID included in the message.
<i>vendorId</i>	The vendor ID included in the message, if any.
<i>messageTag</i>	The tag value that was originally passed by the application in the ::emberRf4ceSend() API.

<i>messageLength</i>	The length in bytes of the message.
<i>message</i>	A pointer to the payload of the message that was sent.

Definition at line 2410 of file [callback.doc](#).

6.30.2.2 void emberRf4ceIncomingMessageHandler (uint8_t *pairingIndex*, uint8_t *profileId*, uint16_t *vendorId*, EmberRf4ceTxOption *txOptions*, uint8_t *messageLength*, uint8_t * *message*)

A callback invoked by the ZigBee RF4CE stack when a message is received.

Parameters

<i>pairingIndex</i>	The index of the entry in the pairing table corresponding to the PAN on which the message was received.
<i>profileId</i>	The profile ID included in the message.
<i>vendorId</i>	The vendor ID included in the message, if any.
<i>txOptions</i>	The TX options used by the source node to transmit the received message.
<i>messageLength</i>	The length in bytes of the received message.
<i>message</i>	A pointer to the payload of the received message.

Definition at line 2438 of file [callback.doc](#).

6.30.2.3 void emberRf4ceDiscoveryCompleteHandler (EmberStatus *status*)

A callback invoked by the ZigBee RF4CE stack when it has completed the discovery process.

Parameters

<i>status</i>	An ::EmberStatus value of ::EMBER_SUCCESS if discovery has been correctly performed over the three RF4CE channels and at least a valid discovery response was received. An ::EmberStatus value of ::EMBER_DISCOVERY_TIMEOUT if the discovery process completed and no valid discovery response was received. Otherwise, another ::EmberStatus value indicating the error occurred.
---------------	--

Definition at line 2459 of file [callback.doc](#).

6.30.2.4 bool emberRf4ceDiscoveryRequestHandler (EmberEUI64 *srcIeeeAddr*, uint8_t *nodeCapabilities*, EmberRf4ceVendorInfo * *vendorInfo*, EmberRf4ceApplicationInfo * *appInfo*, uint8_t *searchDevType*, uint8_t *rxLinkQuality*)

A callback invoked by the ZigBee RF4CE stack when a discovery request is received. If the callback returns true, the stack shall respond with a discovery response, otherwise it will silently discard the discovery request message.

Parameters

<i>srcIeeeAddr</i>	The IEEE address of the node that issued the discovery request.
<i>nodeCapabilities</i>	The node capabilities of the node that issued the discovery request.
<i>vendorInfo</i>	A pointer to an ::EmberRf4ceVendorInfo struct containing the vendor information of the node that issued the discovery request.

<i>appInfo</i>	A pointer to an ::EmberRf4ceApplicationInfo struct containing the application information of the node that issued the discovery request.
<i>searchDevType</i>	The device type being discovered. If this is 0xFF, any type is being requested.
<i>rxLinkQuality</i>	LQI value, as passed via the MAC sub-layer, of the discovery request command frame.

Returns

false if the discovery request should be discarded. Return true if the application wants to respond to the discovery request.

Definition at line 2491 of file [callback.doc](#).

6.30.2.5 bool emberRf4ceDiscoveryResponseHandler (bool *atCapacity*, uint8_t *channel*, EmberPanId *panId*, EmberEUI64 *srcIEEEAddr*, uint8_t *nodeCapabilities*, EmberRf4ceVendorInfo * *vendorInfo*, EmberRf4ceApplicationInfo * *appInfo*, uint8_t *rxLinkQuality*, uint8_t *discRequestLqi*)

A callback invoked by the ZigBee RF4CE stack when a discovery response is received.

Parameters

<i>atCapacity</i>	A bool set to true if the node sending the discovery response has no free entry in its pairing table, false otherwise.
<i>channel</i>	The channel on which the discovery response was received.
<i>panId</i>	The PAN identifier of the responding device.
<i>srcIEEEAddr</i>	The IEEE address of the responding device.
<i>node-Capabilities</i>	The capabilities of the responding node.
<i>vendorInfo</i>	The vendor information of the responding device.
<i>appInfo</i>	The application information of the responding device.
<i>rxLinkQuality</i>	LQI value, as passed via the MAC sub-layer, of the discovery response command frame.
<i>discRequestLqi</i>	The LQI of the discovery request command frame reported by the responding device.

Returns

If this callback returns true the stack will continue the discovery process. If this callback returns false, the discovery process will end at the end of the current discovery trial. A discovery trial is defined as the transmission of a discovery request command frame on all available channels.

Definition at line 2533 of file [callback.doc](#).

6.30.2.6 void emberRf4ceAutoDiscoveryResponseCompleteHandler (EmberStatus *status*, EmberEUI64 *srcIEEEAddr*, uint8_t *nodeCapabilities*, EmberRf4ceVendorInfo * *vendorInfo*, EmberRf4ceApplicationInfo * *appInfo*, uint8_t *searchDevType*)

A callback invoked by the ZigBee RF4CE stack when it has completed the requested auto discovery response phase.

Parameters

<i>status</i>	An ::EmberStatus value of EMBER_SUCCESS indicating that it successfully received a discovery request frame twice from the same node with IEEE address specified by the <i>srcIeeeAddr</i> parameter. An ::EmberStatus value of EMBER_DISCOVERY_TIM-EOUT if the node has not received the two discovery request frame within the auto discovery response duration interval. An ::EmberStatus value of EMBER_DISCOV-ERY_ERROR if the node has received two valid discovery request command frames from two different nodes. An ::EmberStatus value of EMBER_NO_BUFFERS if the node could not respond because of RAM shortage. An ::EmberStatus value of EMBER_ERR_FATAL if the MAC layer rejected the discovery response.
<i>srcIeeeAddr</i>	An ::EmberEUI64 value indicating the IEEE address from which the discovery request command frame was received. This parameter is non-NULL only if the status parameter is EMBER_SUCCESS.
<i>node-Capabilities</i>	The node capabilities of the node that issued the discovery request. This parameter is meaningful only if the status parameter is EMBER_SUCCESS.
<i>vendorInfo</i>	A pointer to an ::EmberRf4ceVendorInfo struct containing the vendor information of the node that issued the discovery request. This parameter is non-NULL only if the status parameter is EMBER_SUCCESS.
<i>appInfo</i>	A pointer to an ::EmberRf4ceApplicationInfo struct containing the application information of the node that issued the discovery request. This parameter is non-NULL only if the status parameter is EMBER_SUCCESS.
<i>searchDevType</i>	The device type being discovered. If this is 0xFF, any type is being requested. This parameter is meaningful only if the status parameter is EMBER_SUCCESS.

Definition at line 2589 of file [callback.doc](#).

6.30.2.7 void emberRf4cePairCompleteHandler (EmberStatus *status*, uint8_t *pairingIndex*, EmberRf4ceVendorInfo * *vendorInfo*, EmberRf4ceApplicationInfo * *applInfo*)

A callback invoked by the ZigBee RF4CE stack when the originator or the recipient node has completed the pairing process.

Parameters

<i>status</i>	An ::EmberStatus value of EMBER_SUCCESS if the pairing process succeeded and a pairing link has been established. An ::EmberStatus value of EMBER_NO_RESPONSE if the originator has timed out waiting for the pair response or for the ping response during the link key exchange procedure. An ::EmberStatus value of EMBER_TABLE_FULL if a pair response was received at the originator indicating that the recipient device has no available entry in its pairing table. An ::EmberStatus value of EMBER_NOT_PERMITTED if a pair response was received at the originator indicating that the recipient device did not accept the pair request. An ::EmberStatus value of EMBER_SECURITY_TIMEOUT if the node has timed out during the link key exchange or recovery procedures. An ::EmberStatus value of EMBER_SECURITY_FAILURE if some other error occurred during the link key exchange or recovery procedures. Another ::EmberStatus value indicating the specific reason why the originator or the recipient node failed to deliver a command frame.
<i>pairingIndex</i>	The index of the pairing table entry corresponding to the pairing link that was established during the pairing process. This field is meaningful only if the status parameter is EMBER_SUCCESS.
<i>vendorInfo</i>	A pointer to an ::EmberRf4ceVendorInfo struct containing the vendor information of the peer node. This parameter is non-NULL only if the status parameter is EMBER_SUCCESS.

<i>appInfo</i>	A pointer to an ::EmberRf4ceApplicationInfo struct containing the application information of the peer node. This parameter is non-NULL only if the status parameter is EMBER_SUCCESS.
----------------	---

Definition at line 2637 of file [callback.doc](#).

6.30.2.8 bool emberRf4cePairRequestHandler (EmberStatus *status*, uint8_t *pairingIndex*, EmberEUI64 *srcIEEEAddr*, uint8_t *nodeCapabilities*, EmberRf4ceVendorInfo * *vendorInfo*, EmberRf4ceApplicationInfo * *appInfo*, uint8_t *keyExchangeTransferCount*)

A callback invoked by the ZigBee RF4CE stack when a pair request has been received.

Parameters

<i>status</i>	An ::EmberStatus value of EMBER_SUCCESS if the request pairing is not a duplicate pairing and an unused entry in the pairing table is available. An ::EmberStatus value of EMBER_TABLE_FULL if the request pairing is not a duplicate pairing and the pairing table is full. An ::EmberStatus value of EMBER_DUPLICATE_ENTRY if the request pairing is a duplicate pairing. In this case, the stack will update the entry indicated by the <i>pairingIndex</i> parameter.
<i>pairingIndex</i>	The index of the entry that will be used by the stack for the pairing link. If the <i>status</i> parameter is EMBER_TABLE_FULL this parameter is meaningless.
<i>srcIEEEAddr</i>	An ::EmberEUI64 value indicating the source IEEE address of the incoming pair request command.
<i>nodeCapabilities</i>	The node capabilities of requesting device.
<i>vendorInfo</i>	The vendor information of the requesting device.
<i>appInfo</i>	The application information of the requesting device.
<i>keyExchangeTransferCount</i>	The number of transfers to be used to exchange the link key with the pairing originator, indicated in the incoming pair request command.

Returns

true if the application accepts the pair, false otherwise.

Definition at line 2677 of file [callback.doc](#).

6.30.2.9 void emberRf4ceUnpairHandler (uint8_t *pairingIndex*)

A callback invoked by the ZigBee RF4CE stack when an unpair command frame has been received. The stack will remove the pairing link indicated by the passed index.

Parameters

<i>pairingIndex</i>	The index of the pairing link to be removed.
---------------------	--

Returns

An ::EmberStatus value of EMBER_SUCCESS if the unpairing process was successfully initiated. Another ::EmberStatus value indicating the specific error occurred otherwise.

Definition at line 2698 of file [callback.doc](#).

6.30.2.10 void emberRf4ceUnpairCompleteHandler (uint8_t *pairingIndex*)

A callback invoked by the ZigBee RF4CE stack when the unpair procedure has been completed. According to the RF4CE specs, during the unpair procedure, the stack sends an unpair command frame. If the command is not successfully delivered, the stack tries another RF4CE channel until the frame is received or the stack already tried all the RF4CE channels. Either way, at the end of the unpair process the pairing table entry is deleted and this callback is invoked.

Parameters

<i>pairingIndex</i>	The index of the removed pairing link.
---------------------	--

Definition at line 2712 of file [callback.doc](#).

6.31 RF4CE Target Communication. Callbacks

Functions

- void [emberAfPluginRf4ceTargetCommunicationHostBindingRequestCallback](#) (void)

6.31.1 Detailed Description

These callbacks are contributed by the RF4CE Target Communication. plugin.

6.31.2 Function Documentation

6.31.2.1 void [emberAfPluginRf4ceTargetCommunicationHostBindingRequestCallback](#) (void)

Host Binding Request.

This function is called by the RF4CE Target Communication plugin when it receives a request from the host to accept incoming binding requests.

Definition at line [2732](#) of file [callback.doc](#).

6.32 ZigBee Remote Control 1.1 Profile Callbacks

Functions

- void `emberAfPluginRf4ceZrc11PairingCompleteCallback` (EmberStatus status, uint8_t pairingIndex, const EmberEUI64 eui64, const EmberRf4ceVendorInfo *vendorInfo, const EmberRf4ceApplicationInfo *applicationInfo)
- void `emberAfPluginRf4ceZrc11UserControlCallback` (const `EmberAfRf4ceZrcUserControlRecord` *record)
- void `emberAfPluginRf4ceZrc11CommandDiscoveryResponseCallback` (EmberStatus status, const `EmberAfRf4ceZrcCommandsSupported` *commandsSupported)

6.32.1 Detailed Description

These callbacks are contributed by the ZigBee Remote Control 1.1 Profile plugin.

6.32.2 Function Documentation

6.32.2.1 void `emberAfPluginRf4ceZrc11PairingCompleteCallback` (`EmberStatus status`, `uint8_t pairingIndex`, `const EmberEUI64 eui64`, `const EmberRf4ceVendorInfo * vendorInfo`, `const EmberRf4ceApplicationInfo * applicationInfo`)

Pairing Complete.

This function is called by the RF4CE ZRC plugin when the push-button pairing operation completes. If status is `:EMBER_SUCCESS`, pairing was successful and pairingIndex indicates the index in the pairing table for the remote node while eui64, vendorInfo, and applicationInfo contain information about the remote node itself. Otherwise, status indicates the reason for failure and the other arguments are meaningless.

Parameters

<code>status</code>	The status of the push-button pairing operation. Ver.: always
<code>pairingIndex</code>	The index of the pairing entry. Ver.: always
<code>eui64</code>	The ::EmberEUI64 of the remote node. Ver.: always
<code>vendorInfo</code>	The vendor information of the remote node. Ver.: always
<code>applicationInfo</code>	The application information of the remote node. Ver.: always

Definition at line 2764 of file `callback.doc`.

6.32.2.2 void `emberAfPluginRf4ceZrc11UserControlCallback` (`const EmberAfRf4ceZrcUserControlRecord * record`)

User Control.

This function is called by the RF4CE ZRC plugin when a user control starts or stops. If the type of the record is `EMBER_AF_RF4CE_ZRC_COMMAND_USER_CONTROL_PRESSED`, the application should execute the requested operation repeatedly at some application-specific rate. When the repetition should stop, the plugin will call the callback again with the type set to `EMBER_AF_RF4CE_ZRC_COMMAND_USER_CONTROL_RELEASED`. `EMBER_AF_RF4CE_ZRC_COMMAND_USER_CONTROL_REPEAT` is a special case of `EMBER_AF_RF4CE_ZRC_COMMAND_USER_CONTROL_PRESSED` and means that the pressed command from the originator was not received and that the originator is still triggering the action. The application should process a repeat type the same as a pressed type, but may wish to perform additional operations to compensate for missed commands.

Parameters

<i>record</i>	The user control record. Ver.: always
---------------	---------------------------------------

Definition at line [2789](#) of file [callback.doc](#).

6.32.2.3 void emberAfPluginRf4ceZrc11CommandDiscoveryResponseCallback (EmberStatus *status*, const EmberAfRf4ceZrcCommandsSupported * *commandsSupported*)

Command Discovery Response.

This function is called by the RF4CE ZRC plugin when a Command Discovery Response message is received by the device.

Parameters

<i>status</i>	An ::EmberStatus value indicating that the command discovery process succeeded or the failure reason. Ver.: always
<i>commands-Supported</i>	The 256-bit field indicating which user control commands are supported. Ver.: always

Definition at line [2803](#) of file [callback.doc](#).

6.33 ZigBee Remote Control 2.0 Profile Callbacks

Functions

- void `emberAfPluginRf4ceZrc20ActionCallback` (const `EmberAfRf4ceZrcActionRecord` *record)
- void `emberAfPluginRf4ceZrc20HaActionCallback` (const `EmberAfRf4ceZrcActionRecord` *record)
- void `emberAfPluginRf4ceZrc20LegacyCommandDiscoveryCompleteCallback` (EmberStatus status, const `EmberAfRf4ceZrcCommandsSupported` *commandsSupported)
- void `emberAfPluginRf4ceZrc20ActionMappingsNegotiationCompleteCallback` (EmberStatus status)
- void `emberAfPluginRf4ceZrc20IncomingMappableActionCallback` (uint8_t pairingIndex, uint16_t entryIndex, `EmberAfRf4ceZrcMappableAction` *mappableAction)
- EmberStatus `emberAfPluginRf4ceZrc20GetMappableActionCallback` (uint8_t pairingIndex, uint16_t entryIndex, `EmberAfRf4ceZrcMappableAction` *mappableAction)
- uint16_t `emberAfPluginRf4ceZrc20GetMappableActionCountCallback` (uint8_t pairingIndex)
- void `emberAfPluginRf4ceZrc20IncomingActionMappingCallback` (uint8_t pairingIndex, uint16_t entryIndex, `EmberAfRf4ceZrcActionMapping` *actionMapping)
- EmberStatus `emberAfPluginRf4ceZrc20GetActionMappingCallback` (uint8_t pairingIndex, uint16_t entryIndex, `EmberAfRf4ceZrcActionMapping` *actionMapping)
- EmberStatus `emberAfPluginRf4ceZrc20SetActionMappingCallback` (uint8_t pairingIndex, uint16_t entryIndex, `EmberAfRf4ceZrcActionMapping` *actionMapping)
- void `emberAfPluginRf4ceZrc20HomeAutomationSupportedAnnouncementCompleteCallback` (EmberStatus status)
- void `emberAfPluginRf4ceZrc20IncomingHomeAutomationSupportedCallback` (uint8_t pairingIndex, uint8_t haInstanceId, `EmberAfRf4ceZrcHomeAutomationSupported` *haSupported)
- EmberStatus `emberAfPluginRf4ceZrc20GetHomeAutomationSupportedCallback` (uint8_t pairingIndex, uint8_t haInstanceId, `EmberAfRf4ceZrcHomeAutomationSupported` *haSupported)
- `EmberAfRf4ceGdpAttributeStatus` `emberAfPluginRf4ceZrc20GetHomeAutomationAttributeCallback` (uint8_t pairingIndex, uint8_t haInstanceId, uint8_t haAttributeId, `EmberAfRf4ceZrcHomeAutomationAttribute` *haAttribute)
- void `emberAfPluginRf4ceZrc20PullHomeAutomationAttributeCompleteCallback` (`EmberAfRf4ceGdpAttributeStatus` responseStatus, `EmberAfRf4ceZrcHomeAutomationAttribute` *haAttribute)

6.33.1 Detailed Description

These callbacks are contributed by the ZigBee Remote Control 2.0 Profile plugin.

6.33.2 Function Documentation

6.33.2.1 void `emberAfPluginRf4ceZrc20ActionCallback` (const `EmberAfRf4ceZrcActionRecord` * record)

Action.

This function is called by the RF4CE ZRC 2.0 plugin when an action starts or stops. If the action type of the action record is `EMBER_AF_RF4CE_ZRC_ACTION_TYPE_START`, the application should execute the requested operation repeatedly at some application-specific rate. When the repetition should stop, the plugin will call the callback again with the action type set to `EMBER_AF_RF4CE_ZRC_ACTION_TYPE_STOP`. `EMBER_AF_RF4CE_ZRC_ACTION_TYPE_REPEAT` is a special case of `EMBER_AF_RF4CE_ZRC_ACTION_TYPE_START` and means that the start action from the originator was not received and that the originator is still triggering the action. The application should process a repeat type the same as a start type, but may wish to perform additional operations to compensate for missed actions. If the action type is `EMBER_AF_RF4CE_ZRC_ACTION_TYPE_ATOMIC`, the application should execute the operation once. The plugin will not call the callback again for an atomic action.

Parameters

<code>record</code>	The action record. Ver.: always
---------------------	---------------------------------

Definition at line 2839 of file [callback.doc](#).

6.33.2.2 void emberAfPluginRf4ceZrc20HaActionCallback (const EmberAfRf4ceZrcActionRecord * *record*)

Ha Action.

This function is called by the RF4CE ZRC 2.0 plugin when an HA action starts or stops. TODO

Parameters

<code>record</code>	The action record. Ver.: always
---------------------	---------------------------------

Definition at line 2850 of file [callback.doc](#).

6.33.2.3 void emberAfPluginRf4ceZrc20LegacyCommandDiscoveryCompleteCallback (EmberStatus *status*, const EmberAfRf4ceZrcCommandsSupported * *commandsSupported*)

Legacy Command Discovery Complete.

This function is called by the RF4CE ZRC plugin upon completion of the legacy ZRC 1.1 command discovery process.

Parameters

<code>status</code>	An ::EmberStatus value indicating that the command discovery process succeeded or the failure reason. Ver.: always
<code>commands-Supported</code>	The 256-bit field indicating which user control commands are supported. Ver.: always

Definition at line 2864 of file [callback.doc](#).

6.33.2.4 void emberAfPluginRf4ceZrc20ActionMappingsNegotiationCompleteCallback (EmberStatus *status*)

Action Mappings Negotiation Complete.

This function is called by the RF4CE ZRC plugin upon completion of the action mapping negotiation procedure.

Parameters

<i>status</i>	An ::EmberStatus value indicating that the action mappings negotiation succeeded or the failure reason. Ver.: always
---------------	--

Definition at line [2877](#) of file [callback.doc](#).

6.33.2.5 void emberAfPluginRf4ceZrc20IncomingMappableActionCallback (uint8_t pairingIndex, uint16_t entryIndex, EmberAfRf4ceZrcMappableAction * mappableAction)

Incoming Mappable Action.

This function is called by an RF4CE ZRC action mapping server upon receiving a mappable action from an already bound action mapping client.

Parameters

<i>pairingIndex</i>	The index of the pairing the mappable action was received from. Ver.: always
<i>entryIndex</i>	The index of the mappable action entry. Ver.: always
<i>mappableAction</i>	A pointer to the mappable action struct that was received. Ver.: always

Definition at line [2892](#) of file [callback.doc](#).

6.33.2.6 EmberStatus emberAfPluginRf4ceZrc20GetMappableActionCallback (uint8_t pairingIndex, uint16_t entryIndex, EmberAfRf4ceZrcMappableAction * mappableAction)

Get Mappable Action.

This function is called by an RF4CE ZRC action mapping client or server to retrieve the mappable action corresponding to the passed entry index.

Parameters

<i>pairingIndex</i>	The index of the pairing the mappable action was received from. Ver.: always
<i>entryIndex</i>	The index of the mappable action entry. Ver.: always
<i>mappableAction</i>	A pointer to the mappable action struct to be populated. Ver.: always

Definition at line [2909](#) of file [callback.doc](#).

6.33.2.7 uint16_t emberAfPluginRf4ceZrc20GetMappableActionCountCallback (uint8_t pairingIndex)

Get Mappable Action Count.

This function is called by an RF4CE ZRC action mapping server or client to retrieve the number of entries currently stored in the mappable action table for a certain pairing.

Parameters

<i>pairingIndex</i>	The index of the pairing entry. Ver.: always
---------------------	--

Definition at line 2924 of file [callback.doc](#).

6.33.2.8 void emberAfPluginRf4ceZrc20IncomingActionMappingCallback (uint8_t *pairingIndex*, uint16_t *entryIndex*, EmberAfRf4ceZrcActionMapping * *actionMapping*)

Incoming Action Mapping.

This function is called by an RF4CE ZRC action mapping client upon receiving an action mapping pulled from an already bound action mapping server.

Parameters

<i>pairingIndex</i>	The index of the pairing the action mapping was received from. Ver.: always
<i>entryIndex</i>	The index of the action mapping entry. Ver.: always
<i>actionMapping</i>	A pointer to the action mapping struct that was received. Ver.: always

Definition at line 2940 of file [callback.doc](#).

6.33.2.9 EmberStatus emberAfPluginRf4ceZrc20GetActionMappingCallback (uint8_t *pairingIndex*, uint16_t *entryIndex*, EmberAfRf4ceZrcActionMapping * *actionMapping*)

Get Action Mapping.

This function is called by an RF4CE ZRC action mapping server to retrieve the action mapping corresponding to the mappable action indicated by the passed entry index.

Parameters

<i>pairingIndex</i>	The index of the pairing the mappable action was received from. Ver.: always
<i>entryIndex</i>	The index of the mappable action entry. Ver.: always
<i>actionMapping</i>	A pointer to the action mapping struct to be populated. Ver.: always

Definition at line 2958 of file [callback.doc](#).

6.33.2.10 EmberStatus emberAfPluginRf4ceZrc20SetActionMappingCallback (uint8_t *pairingIndex*, uint16_t *entryIndex*, EmberAfRf4ceZrcActionMapping * *actionMapping*)

Set Action Mapping.

This function is called by an RF4CE ZRC action mapping server to retrieve the action mapping corresponding to the mappable action indicated by the passed entry index.

Parameters

<i>pairingIndex</i>	The index of the pairing the mappable action was received from. Ver.: always
<i>entryIndex</i>	The index of the mappable action entry. Ver.: always
<i>actionMapping</i>	A pointer to the action mapping struct to be populated. Ver.: always

Definition at line 2977 of file [callback.doc](#).

6.33.2.11 void emberAfPluginRf4ceZrc20HomeAutomationSupportedAnnouncementCompleteCallback (EmberStatus status)

Home Automation Supported Announcement Complete.

This function is called by the RF4CE ZRC plugin upon completion of the Home Automation supported announcement procedure.

Parameters

<i>status</i>	An ::EmberStatus value indicating that the Home Automation supported announcement procedure succeeded or the failure reason. Ver.: always
---------------	---

Definition at line 2993 of file [callback.doc](#).

6.33.2.12 void emberAfPluginRf4ceZrc20IncomingHomeAutomationSupportedCallback (uint8_t pairingIndex, uint8_t haInstanceId, EmberAfRf4ceZrcHomeAutomationSupported * haSupported)

Incoming Home Automation Supported.

This function is called by an RF4CE ZRC Home Automation actions recipient upon receiving a Home Automation supported attribute from an already bound Home Automation originator.

Parameters

<i>pairingIndex</i>	The index of the pairing the Home Automation supported attribute was received from. Ver.: always
<i>haInstanceId</i>	The instance ID the Home Automation supported attribute refers to. Ver.: always
<i>haSupported</i>	A pointer to the Home Automation supported struct that was received. Ver.: always

Definition at line 3010 of file [callback.doc](#).

6.33.2.13 EmberStatus emberAfPluginRf4ceZrc20GetHomeAutomationSupportedCallback (uint8_t pairingIndex, uint8_t haInstanceId, EmberAfRf4ceZrcHomeAutomationSupported * haSupported)

Get Home Automation Supported.

This function is called by an RF4CE ZRC Home Automation actions originator to retrieve the Home Automation supported attribute corresponding to the passed entry index.

Parameters

<i>pairingIndex</i>	The index of the pairing the Home Automation supported attribute was received from. Ver.: always
<i>haInstanceId</i>	The instance ID the Home Automation supported attribute refers to. Ver.: always
<i>haSupported</i>	A pointer to the Home Automation supported struct to be populated. Ver.: always

Definition at line 3029 of file [callback.doc](#).

6.33.2.14 EmberAfRf4ceGdpAttributeStatus emberAfPluginRf4ceZrc20GetHomeAutomationAttributeCallback (uint8_t pairingIndex, uint8_t haInstanceId, uint8_t haAttributelD, EmberAfRf4ceZrcHomeAutomationAttribute * haAttribute)

Get Home Automation Attribute.

This function is called by an RF4CE ZRC Home Automation actions recipient upon receiving a Pull attribute command from a Home Automation actions originator to retrieve the contents of a Home Automation attribute. The contents in the haAttribute struct will be included in the response only if this callback returned an [EmberAfRf4ceGdpAttributeStatus](#) value of [EMBER_AF_RF4CE_GDP_ATTRIBUTE_STATUS_SUCCESS](#).

Parameters

<i>pairingIndex</i>	The index of the pairing entry corresponding to the Home Automation actions originator.. Ver.: always
<i>haInstanceId</i>	The Home Automation instance ID. Ver.: always
<i>haAttributelD</i>	The Home Automation attribute ID. Ver.: always
<i>haAttribute</i>	A pointer to an EmberAfRf4ceZrcHomeAutomationAttribute struct to be populated. Ver.: always

Definition at line 3052 of file [callback.doc](#).

References [EMBER_AF_RF4CE_GDP_ATTRIBUTE_STATUS_UNSUPPORTED_ATTRIBUTE](#).

6.33.2.15 void emberAfPluginRf4ceZrc20PullHomeAutomationAttributeCompleteCallback (EmberAfRf4ceGdpAttributeStatus *responseStatus*, EmberAfRf4ceZrcHomeAutomationAttribute * *haAttribute*)

Pull Home Automation Attribute Complete.

This function is called by an RF4CE ZRC Home Automation actions originator as asynchronous response to a successful call to the [emberAfRf4ceZrc20PullHomeAutomationAttribute\(\)](#) API. This callback is also invoked upon receiving pull attributes responses as result of an initial client notification request for Home Automation attribute pull from the Home Automation actions server.

Parameters

<i>responseStatus</i>	An EmberAfRf4ceGdpAttributeStatus value indication whether the attribute was successfully retrieved or the reason of failure. Ver.: always
<i>haAttribute</i>	A pointer to an EmberAfRf4ceZrcHomeAutomationAttribute struct carrying the contents of the retrieved attribute. This parameter is meaningful only if <i>responseStatus</i> is EMBER_AF_RF4CE_GDP_ATTRIBUTE_STATUS_SUCCESS . Ver.: always

Definition at line 3077 of file [callback.doc](#).

6.34 Stack Minimal Library Callbacks

Functions

- void `emberCounterHandler` (EmberCounterType type, uint8_t data)
- void `emberStackStatusHandler` (EmberStatus status)
- void `emberEnergyScanResultHandler` (uint8_t channel, int8_t maxRssiValue)
- void `emberRadioNeedsCalibratingHandler` (void)

6.34.1 Detailed Description

These callbacks are contributed by the Stack Minimal Library plugin.

6.34.2 Function Documentation

6.34.2.1 void `emberCounterHandler` (EmberCounterType *type*, uint8_t *data*)

A callback invoked to inform the application of the occurrence of an event defined by ::EmberCounterType, for example, transmissions and receptions at different layers of the stack.

The application must define ::EMBER_APPLICATION_HAS_COUNTER_HANDLER in its CONFIGURATION_HEADER to use this. This function may be called in ISR context, so processing should be kept to a minimum.

Parameters

<i>type</i>	The type of the event.
<i>data</i>	For transmission events, the number of retries used. For other events, this parameter is unused and is set to zero.

Definition at line 3106 of file [callback.doc](#).

6.34.2.2 void `emberStackStatusHandler` (EmberStatus *status*)

A callback invoked when the status of the stack changes. If the status parameter equals ::EMBER_NETWORK_UP, then the ::emberGetNetworkParameters() function can be called to obtain the new network parameters. If any of the parameters are being stored in nonvolatile memory by the application, the stored values should be updated.

The application is free to begin messaging once it receives the ::EMBER_NETWORK_UP status. However, routes discovered immediately after the stack comes up may be suboptimal. This is because the routes are based on the neighbor table's information about two-way links with neighboring nodes, which is obtained from periodic ZigBee Link Status messages. It can take two or three link status exchange periods (of 16 seconds each) before the neighbor table has a good estimate of link quality to neighboring nodes. Therefore, the application may improve the quality of initially discovered routes by waiting after startup to give the neighbor table time to be populated.

Parameters

<i>status</i>	Stack status. One of the following: <ul style="list-style-type: none">• ::EMBER_NETWORK_UP• ::EMBER_NETWORK_DOWN• ::EMBER_JOIN_FAILED• ::EMBER_MOVE_FAILED• ::EMBER_CANNOT_JOIN_AS_ROUTER• ::EMBER_NODE_ID_CHANGED• ::EMBER_PAN_ID_CHANGED• ::EMBER_CHANNEL_CHANGED• ::EMBER_NO_BEACONS• ::EMBER RECEIVED_KEY_IN_THE_CLEAR• ::EMBER_NO_NETWORK_KEY_RECEIVED• ::EMBER_NO_LINK_KEY_RECEIVED• ::EMBER_PRECONFIGURED_KEY_REQUIRED
---------------	---

Definition at line 3144 of file [callback.doc](#).

6.34.2.3 void emberEnergyScanResultHandler (uint8_t channel, int8_t maxRssiValue)

Reports the maximum RSSI value measured on the channel.

Parameters

<i>channel</i>	The 802.15.4 channel number on which the RSSI value was measured.
<i>maxRssiValue</i>	The maximum RSSI value measured (in units of dBm).

Definition at line 3156 of file [callback.doc](#).

6.34.2.4 void emberRadioNeedsCalibratingHandler (void)

The radio calibration callback function.

The Voltage Controlled Oscillator (VCO) can drift with temperature changes. During every call to ::emberTick(), the stack will check to see if the VCO has drifted. If the VCO has drifted, the stack will call [emberRadioNeedsCalibratingHandler\(\)](#) to inform the application that it should perform calibration of the current channel as soon as possible. Calibration can take up to 150ms. The default callback function implementation provided here performs calibration immediately. If the application wishes, it can define its own callback by defining ::EMBER_APPLICATION_HAS_CUSTOM_RADIO_CALIBRATION_CALLBACK in its CONFIGURATION_HEADER. It can then failsafe any critical processes or peripherals before calling ::emberCalibrateCurrentChannel(). The application must call ::emberCalibrateCurrentChannel() in response to this callback to maintain expected radio performance.

Definition at line 3176 of file [callback.doc](#).

6.35 STM32F103RET Library Callbacks

Functions

- void [halNcpIsAwakeIsr](#) (bool *isAwake*)

6.35.1 Detailed Description

These callbacks are contributed by the STM32F103RET Library plugin.

6.35.2 Function Documentation

6.35.2.1 void [halNcpIsAwakeIsr](#) (bool *isAwake*)

The SPI Protocol calls [halNcpIsAwakeIsr\(\)](#) once the wakeup handshaking is complete and the NCP is ready to accept a command.

Parameters

<i>isAwake</i>	true if the wake handshake completed and the NCP is awake. false if the wake handshake failed and the NCP is unresponsive.
----------------	--

Definition at line 3200 of file [callback.doc](#).

6.36 sim-eeprom API Callbacks

Functions

- void `halSimEepromCallback` (EmberStatus status)

6.36.1 Detailed Description

These callbacks were contributed by the sim-eeprom API.

6.36.2 Function Documentation

6.36.2.1 void `halSimEepromCallback (EmberStatus status)`

The Simulated EEPROM callback function, implemented by the application.

Parameters

<code>status</code>	An ::EmberStatus error code indicating one of the conditions described below.
---------------------	---

This callback will report an EmberStatus of ::EMBER_SIM_EEPROM_ERASE_PAGE_GREEN whenever a token is set and a page needs to be erased. If the main application loop does not periodically call `halSimEepromErasePage()`, it is best to then erase a page in response to ::EMBER_SIM_EEPROM_ERASE_PAGE_GREEN.

This callback will report an EmberStatus of ::EMBER_SIM_EEPROM_ERASE_PAGE_RED when the pages *must* be erased to prevent data loss. `halSimEepromErasePage()` needs to be called until it returns 0 to indicate there are no more pages that need to be erased. Ignoring this indication and not erasing the pages will cause dropping the new data trying to be written.

This callback will report an EmberStatus of ::EMBER_SIM_EEPROM_FULL when the new data cannot be written due to unerased pages. **Not erasing pages regularly, not erasing in response to ::EMBER_SIM_EEPROM_ERASE_PAGE_GREEN, or not erasing in response to ::EMBER_SIM_EEPROM_ERASE_PAGE_RED will cause ::EMBER_SIM_EEPROM_FULL and the new data will be lost!**. Any future write attempts will be lost as well.

This callback will report an EmberStatus of ::EMBER_SIM_EEPROM_REPAIRING when the Simulated EEPROM needs to repair itself. While there's nothing for an app to do when the SimEE is going to repair itself (SimEE has to be fully functional for the rest of the system to work), alert the application to the fact that repairing is occurring. There are debugging scenarios where an app might want to know that repairing is happening; such as monitoring frequency.

Note

Common situations will trigger an expected repair, such as using a new chip or changing token definitions.

If the callback ever reports the status ::EMBER_ERR_FLASH_WRITE_INHIBITED or ::EMBER_ERR_FLASH_VERIFY_FAILED, this indicates a catastrophic failure in flash writing, meaning either the address being written is not empty or the write itself has failed. If ::EMBER_ERR_FLASH_WRITE_INHIBITED is encountered, the function `:halInternalSimEeRepair(false)` should be called and the chip should then be reset to allow proper initialization to recover. If ::EMBER_ERR_FLASH_VERIFY_FAILED is encountered the Simulated EEPROM (and tokens) on the specific chip with this error should not be trusted anymore.

Definition at line [3262](#) of file [callback.doc](#).

Chapter 7

Data Structure Documentation

7.1 DestStruct Struct Reference

```
#include <rf4ce-zrc20-ha-server.h>
```

Data Fields

- EmberOutgoingMessageType [type](#)
- uint16_t [indexOrDestination](#)
- uint8_t [sourceEndpoint](#)
- uint8_t [destinationEndpoint](#)

7.1.1 Detailed Description

Definition at line 191 of file [rf4ce-zrc20-ha-server.h](#).

The documentation for this struct was generated from the following file:

- [rf4ce-zrc20-ha-server.h](#)

7.2 EmAfBindingInfo Struct Reference

```
#include <rf4ce-gdp-internal.h>
```

Data Fields

- uint8_t [localConfigurationStatus](#)
- uint8_t [candidateIndex](#)

7.2.1 Detailed Description

Definition at line 447 of file [rf4ce-gdp-internal.h](#).

7.2.2 Field Documentation

7.2.2.1 uint8_t EmAfBindingInfo::localConfigurationStatus

Definition at line 448 of file [rf4ce-gdp-internal.h](#).

7.2.2.2 uint8_t EmAfBindingInfo::candidateIndex

Definition at line 449 of file [rf4ce-gdp-internal.h](#).

The documentation for this struct was generated from the following file:

- [rf4ce-gdp-internal.h](#)

7.3 EmAfDiscoveryOrPairRequestData Struct Reference

```
#include <rf4ce-gdp-internal.h>
```

Data Fields

- EmberEUI64 [srcIEEEAddr](#)
- uint8_t [nodeCapabilities](#)
- EmberRf4ceVendorInfo [vendorInfo](#)
- EmberRf4ceApplicationInfo [appInfo](#)
- uint8_t [searchDevType](#)

7.3.1 Detailed Description

Definition at line 439 of file [rf4ce-gdp-internal.h](#).

7.3.2 Field Documentation

7.3.2.1 EmberEUI64 EmAfDiscoveryOrPairRequestData::srcIEEEAddr

Definition at line 440 of file [rf4ce-gdp-internal.h](#).

7.3.2.2 uint8_t EmAfDiscoveryOrPairRequestData::nodeCapabilities

Definition at line 441 of file [rf4ce-gdp-internal.h](#).

7.3.2.3 EmberRf4ceVendorInfo EmAfDiscoveryOrPairRequestData::vendorInfo

Definition at line 442 of file [rf4ce-gdp-internal.h](#).

7.3.2.4 EmberRf4ceApplicationInfo EmAfDiscoveryOrPairRequestData::appInfo

Definition at line 443 of file [rf4ce-gdp-internal.h](#).

7.3.2.5 `uint8_t EmAfDiscoveryOrPairrequestData::searchDevType`

Definition at line 444 of file `rf4ce-gdp-internal.h`.

The documentation for this struct was generated from the following file:

- `rf4ce-gdp-internal.h`

7.4 EmAfGdpPairingCandidatE Struct Reference

```
#include <rf4ce-gdp-internal.h>
```

Data Fields

- `EmberEUI64 ieeeAddr`
- `EmberPanId panId`
- `uint8_t supportedProfiles [EMBER_RF4CE_APPLICATION_PROFILE_ID_LIST_MAX_LENGTH]`
- `uint8_t supportedProfilesLength`
- `uint8_t channel`
- `uint8_t primaryClassDescriptor`
- `uint8_t secondaryClassDescriptor`
- `uint8_t tertiaryClassDescriptor`
- `uint8_t rxLqi`
- `uint8_t info`

7.4.1 Detailed Description

Definition at line 426 of file `rf4ce-gdp-internal.h`.

7.4.2 Field Documentation

7.4.2.1 `EmberEUI64 EmAfGdpPairingCandidatE::ieeeAddr`

Definition at line 427 of file `rf4ce-gdp-internal.h`.

7.4.2.2 `EmberPanId EmAfGdpPairingCandidatE::panId`

Definition at line 428 of file `rf4ce-gdp-internal.h`.

7.4.2.3 `uint8_t EmAfGdpPairingCandidatE::supportedProfiles[EMBER_RF4CE_APPLICATION_PROFILE_ID_LIST_MAX_LENGTH]`

Definition at line 429 of file `rf4ce-gdp-internal.h`.

7.4.2.4 `uint8_t EmAfGdpPairingCandidatE::supportedProfilesLength`

Definition at line 430 of file `rf4ce-gdp-internal.h`.

7.4.2.5 `uint8_t EmAfGdpPairingCandidat::channel`

Definition at line 431 of file [rf4ce-gdp-internal.h](#).

7.4.2.6 `uint8_t EmAfGdpPairingCandidat::primaryClassDescriptor`

Definition at line 432 of file [rf4ce-gdp-internal.h](#).

7.4.2.7 `uint8_t EmAfGdpPairingCandidat::secondaryClassDescriptor`

Definition at line 433 of file [rf4ce-gdp-internal.h](#).

7.4.2.8 `uint8_t EmAfGdpPairingCandidat::tertiaryClassDescriptor`

Definition at line 434 of file [rf4ce-gdp-internal.h](#).

7.4.2.9 `uint8_t EmAfGdpPairingCandidat::rxLqi`

Definition at line 435 of file [rf4ce-gdp-internal.h](#).

7.4.2.10 `uint8_t EmAfGdpPairingCandidat::info`

Definition at line 436 of file [rf4ce-gdp-internal.h](#).

The documentation for this struct was generated from the following file:

- [rf4ce-gdp-internal.h](#)

7.5 EmAfMsoPairingCandidat Struct Reference

```
#include <rf4ce-mso-internal.h>
```

Data Fields

- `uint8_t channel`
- `EmberPanId panId`
- `EmberEUI64 ieeeAddr`
- `uint8_t primaryClassDescriptor`
- `uint8_t secondaryClassDescriptor`
- `uint8_t tertiaryClassDescriptor`
- `uint8_t basicLqiThreshold`
- `uint8_t strictLqiThreshold`
- `uint8_t rxLqi`
- `uint8_t control`

7.5.1 Detailed Description

Definition at line 256 of file [rf4ce-mso-internal.h](#).

7.5.2 Field Documentation

7.5.2.1 `uint8_t EmAfMsoPairingCandidat::channel`

Definition at line 257 of file [rf4ce-mso-internal.h](#).

7.5.2.2 `EmberPanId EmAfMsoPairingCandidat::panId`

Definition at line 258 of file [rf4ce-mso-internal.h](#).

7.5.2.3 `EmberEUI64 EmAfMsoPairingCandidat::ieeeAddr`

Definition at line 259 of file [rf4ce-mso-internal.h](#).

7.5.2.4 `uint8_t EmAfMsoPairingCandidat::primaryClassDescriptor`

Definition at line 260 of file [rf4ce-mso-internal.h](#).

7.5.2.5 `uint8_t EmAfMsoPairingCandidat::secondaryClassDescriptor`

Definition at line 261 of file [rf4ce-mso-internal.h](#).

7.5.2.6 `uint8_t EmAfMsoPairingCandidat::tertiaryClassDescriptor`

Definition at line 262 of file [rf4ce-mso-internal.h](#).

7.5.2.7 `uint8_t EmAfMsoPairingCandidat::basicLqiThreshold`

Definition at line 263 of file [rf4ce-mso-internal.h](#).

7.5.2.8 `uint8_t EmAfMsoPairingCandidat::strictLqiThreshold`

Definition at line 264 of file [rf4ce-mso-internal.h](#).

7.5.2.9 `uint8_t EmAfMsoPairingCandidat::rxLqi`

Definition at line 265 of file [rf4ce-mso-internal.h](#).

7.5.2.10 `uint8_t EmAfMsoPairingCandidat::control`

Definition at line 266 of file [rf4ce-mso-internal.h](#).

The documentation for this struct was generated from the following file:

- [rf4ce-mso-internal.h](#)

7.6 EmAfRf4ceGdpAttributeDescriptor Struct Reference

```
#include <rf4ce-gdp-attributes.h>
```

Data Fields

- `uint8_t id`
- `uint8_t size`
- `uint8_t bitmask`
- `uint16_t dimension`

7.6.1 Detailed Description

Definition at line 131 of file [rf4ce-gdp-attributes.h](#).

7.6.2 Field Documentation

7.6.2.1 `uint8_t EmAfRf4ceGdpAttributeDescriptor::id`

Definition at line 132 of file [rf4ce-gdp-attributes.h](#).

7.6.2.2 `uint8_t EmAfRf4ceGdpAttributeDescriptor::size`

Definition at line 133 of file [rf4ce-gdp-attributes.h](#).

7.6.2.3 `uint8_t EmAfRf4ceGdpAttributeDescriptor::bitmask`

Definition at line 134 of file [rf4ce-gdp-attributes.h](#).

7.6.2.4 `uint16_t EmAfRf4ceGdpAttributeDescriptor::dimension`

Definition at line 135 of file [rf4ce-gdp-attributes.h](#).

The documentation for this struct was generated from the following file:

- [rf4ce-gdp-attributes.h](#)

7.7 EmAfRf4ceGdpAttributes Struct Reference

```
#include <rf4ce-gdp-attributes.h>
```

Data Fields

- `uint16_t gdpVersion`
- `uint32_t gdpCapabilities`
- `uint8_t powerStatus`
- `uint8_t pollConstraints [APL_GDP_POLL_CONSTRAINTS_SIZE]`
- `uint8_t pollConfiguration [APL_GDP_POLL_CONFIGURATION_SIZE]`
- `uint16_t autoCheckValidationPeriod`
- `uint16_t linkLostWaitTime`
- `uint8_t identificationCapabilities`

7.7.1 Detailed Description

Definition at line 168 of file `rf4ce-gdp-attributes.h`.

7.7.2 Field Documentation

7.7.2.1 `uint16_t EmAfRf4ceGdpAttributes::gdpVersion`

Definition at line 169 of file `rf4ce-gdp-attributes.h`.

7.7.2.2 `uint32_t EmAfRf4ceGdpAttributes::gdpCapabilities`

Definition at line 170 of file `rf4ce-gdp-attributes.h`.

7.7.2.3 `uint8_t EmAfRf4ceGdpAttributes::powerStatus`

Definition at line 171 of file `rf4ce-gdp-attributes.h`.

7.7.2.4 `uint8_t EmAfRf4ceGdpAttributes::pollConstraints[APL_GDP_POLL_CONSTRAINTS_SIZE]`

Definition at line 172 of file `rf4ce-gdp-attributes.h`.

7.7.2.5 `uint8_t EmAfRf4ceGdpAttributes::pollConfiguration[APL_GDP_POLL_CONFIGURATION_SIZE]`

Definition at line 173 of file `rf4ce-gdp-attributes.h`.

7.7.2.6 `uint16_t EmAfRf4ceGdpAttributes::autoCheckValidationPeriod`

Definition at line 174 of file `rf4ce-gdp-attributes.h`.

7.7.2.7 `uint16_t EmAfRf4ceGdpAttributes::linkLostWaitTime`

Definition at line 175 of file `rf4ce-gdp-attributes.h`.

7.7.2.8 `uint8_t EmAfRf4ceGdpAttributes::identificationCapabilities`

Definition at line 176 of file [rf4ce-gdp-attributes.h](#).

The documentation for this struct was generated from the following file:

- [rf4ce-gdp-attributes.h](#)

7.8 `EmAfRf4ceMsoAttributeDescriptor` Struct Reference

```
#include <rf4ce-mso-attributes.h>
```

Data Fields

- [EmberAfRf4ceMsoAttributeId id](#)
- `uint8_t size`
- `uint8_t bitmask`
- `uint8_t dimension`

7.8.1 Detailed Description

Definition at line 29 of file [rf4ce-mso-attributes.h](#).

7.8.2 Field Documentation

7.8.2.1 `EmberAfRf4ceMsoAttributeId EmAfRf4ceMsoAttributeDescriptor::id`

Definition at line 30 of file [rf4ce-mso-attributes.h](#).

7.8.2.2 `uint8_t EmAfRf4ceMsoAttributeDescriptor::size`

Definition at line 31 of file [rf4ce-mso-attributes.h](#).

7.8.2.3 `uint8_t EmAfRf4ceMsoAttributeDescriptor::bitmask`

Definition at line 32 of file [rf4ce-mso-attributes.h](#).

7.8.2.4 `uint8_t EmAfRf4ceMsoAttributeDescriptor::dimension`

Definition at line 33 of file [rf4ce-mso-attributes.h](#).

The documentation for this struct was generated from the following file:

- [rf4ce-mso-attributes.h](#)

7.9 EmAfRf4ceMsoPeripheralIdEntry Struct Reference

```
#include <rf4ce-mso-attributes.h>
```

Data Fields

- uint8_t `deviceType`
- uint8_t `peripheralId` [MSO_RIB_ATTRIBUTE_PERIPHERAL_IDS_LENGTH]

7.9.1 Detailed Description

Definition at line 36 of file [rf4ce-mso-attributes.h](#).

7.9.2 Field Documentation

7.9.2.1 uint8_t EmAfRf4ceMsoPeripheralIdEntry::deviceType

Definition at line 37 of file [rf4ce-mso-attributes.h](#).

7.9.2.2 uint8_t EmAfRf4ceMsoPeripheralIdEntry::peripheralId[MSO_RIB_ATTRIBUTE_PERIPHERAL_IDS_LENGTH]

Definition at line 38 of file [rf4ce-mso-attributes.h](#).

The documentation for this struct was generated from the following file:

- [rf4ce-mso-attributes.h](#)

7.10 EmAfRf4ceMsoRibAttributes Struct Reference

```
#include <rf4ce-mso-attributes.h>
```

Data Fields

- `EmAfRf4ceMsoPeripheralIdEntry peripheralIds` [EMBER_AF_PLUGIN_RF4CE_MSO_PERIPHERAL_ID_ENTRIES]
- uint8_t `rfStatistics` [MSO_RIB_ATTRIBUTE_RF_STATISTICS_LENGTH]
- uint8_t `versioning` [MSO_ATTRIBUTE_VERSIONING_ENTRIES][MSO_RIB_ATTRIBUTE_VERSIONING_LENGTH]
- uint8_t `batteryStatus` [MSO_RIB_ATTRIBUTE_BATTERY_STATUS_LENGTH]
- uint8_t `shortRfRetryPeriod` [MSO_RIB_ATTRIBUTE_SHORT_RF_RETRY_PERIOD_LENGTH]
- uint8_t `validationConfiguration` [MSO_RIB_ATTRIBUTE_VALIDATION_CONFIGURATION_LENGTH]
- uint8_t `generalPurpose` [EMBER_AF_PLUGIN_RF4CE_MSO_GENERAL_PURPOSE_ENTRIES][MSO_RIB_ATTRIBUTE_GENERAL_PURPOSE_LENGTH]

7.10.1 Detailed Description

Definition at line 41 of file [rf4ce-mso-attributes.h](#).

7.10.2 Field Documentation

7.10.2.1 `EmAfRf4ceMsoPeripheralIdEntry EmAfRf4ceMsoRibAttributes::peripheralIds[EMBER_AF_PLUGIN_RF4CE_MSO_PERIPHERAL_ID_ENTRIES]`

Definition at line 42 of file [rf4ce-mso-attributes.h](#).

7.10.2.2 `uint8_t EmAfRf4ceMsoRibAttributes::rfStatistics[MSO_RIB_ATTRIBUTE_RF_STATISTICS_LENGTH]`

Definition at line 43 of file [rf4ce-mso-attributes.h](#).

7.10.2.3 `uint8_t EmAfRf4ceMsoRibAttributes::versioning[MSO_ATTRIBUTE_VERSIONING_ENTRIES][MSO_RIB_ATTRIBUTE_VERSIONING_LENGTH]`

Definition at line 44 of file [rf4ce-mso-attributes.h](#).

7.10.2.4 `uint8_t EmAfRf4ceMsoRibAttributes::batteryStatus[MSO_RIB_ATTRIBUTE_BATTERY_STATUS_LENGTH]`

Definition at line 45 of file [rf4ce-mso-attributes.h](#).

7.10.2.5 `uint8_t EmAfRf4ceMsoRibAttributes::shortRfRetryPeriod[MSO_RIB_ATTRIBUTE_SHORT_RF_RETRY_PERIOD_LENGTH]`

Definition at line 46 of file [rf4ce-mso-attributes.h](#).

7.10.2.6 `uint8_t EmAfRf4ceMsoRibAttributes::validationConfiguration[MSO_RIB_ATTRIBUTE_VALIDATION_CONFIGURATION_LENGTH]`

Definition at line 48 of file [rf4ce-mso-attributes.h](#).

7.10.2.7 `uint8_t EmAfRf4ceMsoRibAttributes::generalPurpose[EMBER_AF_PLUGIN_RF4CE_MSO_GENERAL_PURPOSE_ENTRIES][MSO_RIB_ATTRIBUTE_GENERAL_PURPOSE_LENGTH]`

Definition at line 49 of file [rf4ce-mso-attributes.h](#).

The documentation for this struct was generated from the following file:

- [rf4ce-mso-attributes.h](#)

7.11 EmAfRf4cePowerSavingState Struct Reference

```
#include <rf4ce-profile-internal.h>
```

Data Fields

- uint32_t [dutyCycleMs](#)
- uint32_t [activePeriodMs](#)

7.11.1 Detailed Description

Definition at line [108](#) of file [rf4ce-profile-internal.h](#).

7.11.2 Field Documentation

7.11.2.1 uint32_t EmAfRf4cePowerSavingState::dutyCycleMs

Definition at line [109](#) of file [rf4ce-profile-internal.h](#).

7.11.2.2 uint32_t EmAfRf4cePowerSavingState::activePeriodMs

Definition at line [110](#) of file [rf4ce-profile-internal.h](#).

The documentation for this struct was generated from the following file:

- [rf4ce-profile-internal.h](#)

7.12 EmAfRf4ceZrcAttributeDescriptor Struct Reference

```
#include <rf4ce-zrc20-attributes.h>
```

Data Fields

- uint8_t [id](#)
- uint8_t [size](#)
- uint8_t [bitmask](#)

7.12.1 Detailed Description

Definition at line [68](#) of file [rf4ce-zrc20-attributes.h](#).

7.12.2 Field Documentation

7.12.2.1 uint8_t EmAfRf4ceZrcAttributeDescriptor::id

Definition at line [69](#) of file [rf4ce-zrc20-attributes.h](#).

7.12.2.2 `uint8_t EmAfRf4ceZrcAttributeDescriptor::size`

Definition at line 70 of file [rf4ce-zrc20-attributes.h](#).

7.12.2.3 `uint8_t EmAfRf4ceZrcAttributeDescriptor::bitmask`

Definition at line 71 of file [rf4ce-zrc20-attributes.h](#).

The documentation for this struct was generated from the following file:

- [rf4ce-zrc20-attributes.h](#)

7.13 EmAfRf4ceZrcAttributes Struct Reference

```
#include <rf4ce-zrc20-attributes.h>
```

Data Fields

- `uint16_t zrcProfileVersion`
- `uint16_t zrcActionBanksVersion`
- `EmAfZrcBitmask * actionBanksSupportedRx`
- `EmAfZrcBitmask * actionBanksSupportedTx`
- `EmAfZrcArrayedBitmask * actionCodesSupportedRx`
- `EmAfZrcArrayedBitmask * actionCodesSupportedTx`

7.13.1 Detailed Description

Definition at line 84 of file [rf4ce-zrc20-attributes.h](#).

7.13.2 Field Documentation

7.13.2.1 `uint16_t EmAfRf4ceZrcAttributes::zrcProfileVersion`

Definition at line 86 of file [rf4ce-zrc20-attributes.h](#).

7.13.2.2 `uint16_t EmAfRf4ceZrcAttributes::zrcActionBanksVersion`

Definition at line 87 of file [rf4ce-zrc20-attributes.h](#).

7.13.2.3 `EmAfZrcBitmask* EmAfRf4ceZrcAttributes::actionBanksSupportedRx`

Definition at line 88 of file [rf4ce-zrc20-attributes.h](#).

7.13.2.4 `EmAfZrcBitmask* EmAfRf4ceZrcAttributes::actionBanksSupportedTx`

Definition at line 89 of file [rf4ce-zrc20-attributes.h](#).

7.13.2.5 EmAfZrcArrayedBitmask* EmAfRf4ceZrcAttributes::actionCodesSupportedRx

Definition at line 97 of file [rf4ce-zrc20-attributes.h](#).

7.13.2.6 EmAfZrcArrayedBitmask* EmAfRf4ceZrcAttributes::actionCodesSupportedTx

Definition at line 98 of file [rf4ce-zrc20-attributes.h](#).

The documentation for this struct was generated from the following file:

- [rf4ce-zrc20-attributes.h](#)

7.14 EmAfZrcArrayedBitmask Struct Reference

```
#include <rf4ce-zrc20-attributes.h>
```

Data Fields

- bool [inUse](#)
- uint8_t [entryId](#)
- uint8_t [contents](#) [ZRC_BITMASK_SIZE]

7.14.1 Detailed Description

Definition at line 78 of file [rf4ce-zrc20-attributes.h](#).

7.14.2 Field Documentation

7.14.2.1 bool EmAfZrcArrayedBitmask::inUse

Definition at line 79 of file [rf4ce-zrc20-attributes.h](#).

7.14.2.2 uint8_t EmAfZrcArrayedBitmask::entryId

Definition at line 80 of file [rf4ce-zrc20-attributes.h](#).

7.14.2.3 uint8_t EmAfZrcArrayedBitmask::contents[ZRC_BITMASK_SIZE]

Definition at line 81 of file [rf4ce-zrc20-attributes.h](#).

The documentation for this struct was generated from the following file:

- [rf4ce-zrc20-attributes.h](#)

7.15 EmAfZrcBitmask Struct Reference

```
#include <rf4ce-zrc20-attributes.h>
```

Data Fields

- `uint8_t contents [ZRC_BITMASK_SIZE]`

7.15.1 Detailed Description

Definition at line 74 of file [rf4ce-zrc20-attributes.h](#).

7.15.2 Field Documentation

7.15.2.1 `uint8_t EmAfZrcBitmask::contents[ZRC_BITMASK_SIZE]`

Definition at line 75 of file [rf4ce-zrc20-attributes.h](#).

The documentation for this struct was generated from the following file:

- [rf4ce-zrc20-attributes.h](#)

7.16 EmberAfRf4ceGdpAttributeldentificationRecord Struct Reference

```
#include <rf4ce-gdp-types.h>
```

Data Fields

- `EmberAfRf4ceGdpAttributeId attributeId`
- `uint16_t entryId`

7.16.1 Detailed Description

RF4CE GDP attribute identification record for Get Attributes and Pull Attributes messages.

Definition at line 135 of file [rf4ce-gdp-types.h](#).

7.16.2 Field Documentation

7.16.2.1 `EmberAfRf4ceGdpAttributeId EmberAfRf4ceGdpAttributeldentificationRecord::attributeld`

Definition at line 136 of file [rf4ce-gdp-types.h](#).

7.16.2.2 `uint16_t EmberAfRf4ceGdpAttributeldentificationRecord::entryId`

Definition at line 137 of file [rf4ce-gdp-types.h](#).

The documentation for this struct was generated from the following file:

- [rf4ce-gdp-types.h](#)

7.17 EmberAfRf4ceGdpAttributeRecord Struct Reference

```
#include <rf4ce-gdp-types.h>
```

Data Fields

- `EmberAfRf4ceGdpAttributeId attributeId`
- `uint16_t entryId`
- `uint8_t valueLength`
- `const uint8_t * value`

7.17.1 Detailed Description

RF4CE GDP attribute identification record for Set Attributes and Push Attributes messages.

Definition at line 156 of file `rf4ce-gdp-types.h`.

7.17.2 Field Documentation

7.17.2.1 EmberAfRf4ceGdpAttributeId EmberAfRf4ceGdpAttributeRecord::attributeId

Definition at line 157 of file `rf4ce-gdp-types.h`.

7.17.2.2 uint16_t EmberAfRf4ceGdpAttributeRecord::entryId

Definition at line 158 of file `rf4ce-gdp-types.h`.

7.17.2.3 uint8_t EmberAfRf4ceGdpAttributeRecord::valueLength

Definition at line 159 of file `rf4ce-gdp-types.h`.

7.17.2.4 const uint8_t* EmberAfRf4ceGdpAttributeRecord::value

Definition at line 160 of file `rf4ce-gdp-types.h`.

The documentation for this struct was generated from the following file:

- `rf4ce-gdp-types.h`

7.18 EmberAfRf4ceGdpAttributeStatusRecord Struct Reference

```
#include <rf4ce-gdp-types.h>
```

Data Fields

- `EmberAfRf4ceGdpAttributeId attributeId`
- `uint16_t entryId`

- `EmberAfRf4ceGdpAttributeStatus status`
- `uint8_t valueLength`
- `const uint8_t * value`

7.18.1 Detailed Description

RF4CE GDP attribute identification record for Get Attributes Response and Pull Attributes Response messages.

Definition at line 144 of file `rf4ce-gdp-types.h`.

7.18.2 Field Documentation

7.18.2.1 `EmberAfRf4ceGdpAttributeId EmberAfRf4ceGdpAttributeStatusRecord::attributelid`

Definition at line 145 of file `rf4ce-gdp-types.h`.

7.18.2.2 `uint16_t EmberAfRf4ceGdpAttributeStatusRecord::entryId`

Definition at line 146 of file `rf4ce-gdp-types.h`.

7.18.2.3 `EmberAfRf4ceGdpAttributeStatus EmberAfRf4ceGdpAttributeStatusRecord::status`

Definition at line 147 of file `rf4ce-gdp-types.h`.

7.18.2.4 `uint8_t EmberAfRf4ceGdpAttributeStatusRecord::valueLength`

Definition at line 148 of file `rf4ce-gdp-types.h`.

7.18.2.5 `const uint8_t* EmberAfRf4ceGdpAttributeStatusRecord::value`

Definition at line 149 of file `rf4ce-gdp-types.h`.

The documentation for this struct was generated from the following file:

- `rf4ce-gdp-types.h`

7.19 `EmberAfRf4ceGdpRand` Struct Reference

```
#include <rf4ce-gdp-types.h>
```

Data Fields

- `uint8_t contents [EMBER_AF_RF4CE_GDP_RAND_SIZE]`

7.19.1 Detailed Description

This data structure contains the GDP random byte string that is passed into various other functions.

Definition at line 265 of file [rf4ce-gdp-types.h](#).

7.19.2 Field Documentation

7.19.2.1 `uint8_t EmberAfRf4ceGdpRand::contents[EMBER_AF_RF4CE_GDP_RAND_SIZE]`

This is the random byte string.

Definition at line 267 of file [rf4ce-gdp-types.h](#).

The documentation for this struct was generated from the following file:

- [rf4ce-gdp-types.h](#)

7.20 EmberAfRf4ceGdpTag Struct Reference

```
#include <rf4ce-gdp-types.h>
```

Data Fields

- `uint8_t contents [EMBER_AF_RF4CE_GDP_TAG_SIZE]`

7.20.1 Detailed Description

This data structure contains the GDP tag value that is passed into various other functions.

Definition at line 279 of file [rf4ce-gdp-types.h](#).

7.20.2 Field Documentation

7.20.2.1 `uint8_t EmberAfRf4ceGdpTag::contents[EMBER_AF_RF4CE_GDP_TAG_SIZE]`

This is the tag value.

Definition at line 281 of file [rf4ce-gdp-types.h](#).

The documentation for this struct was generated from the following file:

- [rf4ce-gdp-types.h](#)

7.21 EmberAfRf4ceMsolrRfDatabaseEntry Struct Reference

```
#include <rf4ce-mso-types.h>
```

Data Fields

- `EmberAfRf4ceMsoIrRfDatabaseFlags` flags
- `EmberAfRf4ceMsoIrRfDatabaseRfDescriptor` rfPressedDescriptor
- `EmberAfRf4ceMsoIrRfDatabaseRfDescriptor` rfRepeatedDescriptor
- `EmberAfRf4ceMsoIrRfDatabaseRfDescriptor` rfReleasedDescriptor
- `EmberAfRf4ceMsoIrRfDatabaseIrDescriptor` irDescriptor

7.21.1 Detailed Description

RF4CE MSO IR-RF database entry.

Definition at line 472 of file `rf4ce-mso-types.h`.

7.21.2 Field Documentation

7.21.2.1 EmberAfRf4ceMsoIrRfDatabaseFlags EmberAfRf4ceMsolRfDatabaseEntry::flags

Definition at line 473 of file `rf4ce-mso-types.h`.

7.21.2.2 EmberAfRf4ceMsoIrRfDatabaseRfDescriptor EmberAfRf4ceMsolRfDatabaseEntry::rf-PressedDescriptor

Definition at line 474 of file `rf4ce-mso-types.h`.

7.21.2.3 EmberAfRf4ceMsoIrRfDatabaseRfDescriptor EmberAfRf4ceMsolRfDatabaseEntry::rf-RepeatedDescriptor

Definition at line 475 of file `rf4ce-mso-types.h`.

7.21.2.4 EmberAfRf4ceMsoIrRfDatabaseRfDescriptor EmberAfRf4ceMsolRfDatabaseEntry::rf-ReleasedDescriptor

Definition at line 476 of file `rf4ce-mso-types.h`.

7.21.2.5 EmberAfRf4ceMsoIrRfDatabaseIrDescriptor EmberAfRf4ceMsolRfDatabaseEntry::ir-Descriptor

Definition at line 477 of file `rf4ce-mso-types.h`.

The documentation for this struct was generated from the following file:

- `rf4ce-mso-types.h`

7.22 EmberAfRf4ceMsolRfDatabaseIrDescriptor Struct Reference

```
#include <rf4ce-mso-types.h>
```

Data Fields

- `EmberAfRf4ceMsoIrRfDatabaseIrConfig irConfig`
- `uint8_t irCodeLength`
- `const uint8_t * irCode`

7.22.1 Detailed Description

RF4CE MSO IR-RF database IR descriptor.

Definition at line 463 of file `rf4ce-mso-types.h`.

7.22.2 Field Documentation

7.22.2.1 `EmberAfRf4ceMsoIrRfDatabaseIrConfig EmberAfRf4ceMsolrRfDatabaseIrDescriptor::irConfig`

Definition at line 464 of file `rf4ce-mso-types.h`.

7.22.2.2 `uint8_t EmberAfRf4ceMsolrRfDatabaseIrDescriptor::irCodeLength`

Definition at line 465 of file `rf4ce-mso-types.h`.

7.22.2.3 `const uint8_t* EmberAfRf4ceMsolrRfDatabaseIrDescriptor::irCode`

Definition at line 466 of file `rf4ce-mso-types.h`.

The documentation for this struct was generated from the following file:

- `rf4ce-mso-types.h`

7.23 `EmberAfRf4ceMsolrRfDatabaseRfDescriptor` Struct Reference

```
#include <rf4ce-mso-types.h>
```

Data Fields

- `EmberAfRf4ceMsoIrRfDatabaseRfConfig rfConfig`
- `EmberRf4ceTxOption txOptions`
- `uint8_t payloadLength`
- `const uint8_t * payload`

7.23.1 Detailed Description

RF4CE MSO IR-RF database RF descriptor.

Definition at line 423 of file `rf4ce-mso-types.h`.

7.23.2 Field Documentation

7.23.2.1 EmberAfRf4ceMsoIrRfDatabaseRfConfig EmberAfRf4ceMsoIrRfDatabaseRfDescriptor::rfConfig

Definition at line 424 of file [rf4ce-mso-types.h](#).

7.23.2.2 EmberRf4ceTxOption EmberAfRf4ceMsoIrRfDatabaseRfDescriptor::txOptions

Definition at line 425 of file [rf4ce-mso-types.h](#).

7.23.2.3 uint8_t EmberAfRf4ceMsoIrRfDatabaseRfDescriptor::payloadLength

Definition at line 426 of file [rf4ce-mso-types.h](#).

7.23.2.4 const uint8_t* EmberAfRf4ceMsoIrRfDatabaseRfDescriptor::payload

Definition at line 427 of file [rf4ce-mso-types.h](#).

The documentation for this struct was generated from the following file:

- [rf4ce-mso-types.h](#)

7.24 EmberAfRf4ceMsoUserControlRecord Struct Reference

```
#include <rf4ce-mso-types.h>
```

Data Fields

- uint8_t pairingIndex
- [EmberAfRf4ceMsoCommandCode](#) commandCode
- [EmberAfRf4ceMsoKeyCode](#) rcCommandCode
- const uint8_t * rcCommandPayload
- uint8_t rcCommandPayloadLength
- uint16_t timeMs

7.24.1 Detailed Description

This data structure contains the MSO user control record.

Definition at line 320 of file [rf4ce-mso-types.h](#).

7.24.2 Field Documentation

7.24.2.1 uint8_t EmberAfRf4ceMsoUserControlRecord::pairingIndex

Definition at line 321 of file [rf4ce-mso-types.h](#).

7.24.2.2 EmberAfRf4ceMsoCommandCode EmberAfRf4ceMsoUserControlRecord::commandCode

Definition at line 322 of file [rf4ce-mso-types.h](#).

7.24.2.3 EmberAfRf4ceMsoKeyCode EmberAfRf4ceMsoUserControlRecord::rcCommandCode

Definition at line 323 of file [rf4ce-mso-types.h](#).

7.24.2.4 const uint8_t* EmberAfRf4ceMsoUserControlRecord::rcCommandPayload

Definition at line 324 of file [rf4ce-mso-types.h](#).

7.24.2.5 uint8_t EmberAfRf4ceMsoUserControlRecord::rcCommandPayloadLength

Definition at line 325 of file [rf4ce-mso-types.h](#).

7.24.2.6 uint16_t EmberAfRf4ceMsoUserControlRecord::timeMs

Definition at line 326 of file [rf4ce-mso-types.h](#).

The documentation for this struct was generated from the following file:

- [rf4ce-mso-types.h](#)

7.25 EmberAfRf4ceZrcActionMapping Struct Reference

```
#include <rf4ce-zrc20-types.h>
```

Data Fields

- `uint8_t mappingFlags`
- `uint8_t rfConfig`
- `uint8_t rf4ceTxOptions`
- `uint8_t actionDataLength`
- `uint8_t * actionData`
- `uint8_t irConfig`
- `uint16_t irVendorId`
- `uint8_t irCodeLength`
- `uint8_t * irCode`

7.25.1 Detailed Description

RF4CE ZRC Action Mapping.

Definition at line 403 of file [rf4ce-zrc20-types.h](#).

7.25.2 Field Documentation

7.25.2.1 `uint8_t EmberAfRf4ceZrcActionMapping::mappingFlags`

Definition at line 404 of file [rf4ce-zrc20-types.h](#).

7.25.2.2 `uint8_t EmberAfRf4ceZrcActionMapping::rfConfig`

Definition at line 406 of file [rf4ce-zrc20-types.h](#).

7.25.2.3 `uint8_t EmberAfRf4ceZrcActionMapping::rf4ceTxOptions`

Definition at line 407 of file [rf4ce-zrc20-types.h](#).

7.25.2.4 `uint8_t EmberAfRf4ceZrcActionMapping::actionDataLength`

Definition at line 408 of file [rf4ce-zrc20-types.h](#).

7.25.2.5 `uint8_t* EmberAfRf4ceZrcActionMapping::actionData`

Definition at line 409 of file [rf4ce-zrc20-types.h](#).

7.25.2.6 `uint8_t EmberAfRf4ceZrcActionMapping::irConfig`

Definition at line 411 of file [rf4ce-zrc20-types.h](#).

7.25.2.7 `uint16_t EmberAfRf4ceZrcActionMapping::irVendorId`

Definition at line 412 of file [rf4ce-zrc20-types.h](#).

7.25.2.8 `uint8_t EmberAfRf4ceZrcActionMapping::irCodeLength`

Definition at line 413 of file [rf4ce-zrc20-types.h](#).

7.25.2.9 `uint8_t* EmberAfRf4ceZrcActionMapping::irCode`

Definition at line 414 of file [rf4ce-zrc20-types.h](#).

The documentation for this struct was generated from the following file:

- [rf4ce-zrc20-types.h](#)

7.26 EmberAfRf4ceZrcActionRecord Struct Reference

```
#include <rf4ce-zrc20-types.h>
```

Data Fields

- `uint8_t pairingIndex`
- `EmberAfRf4ceZrcActionType actionType`
- `EmberAfRf4ceZrcModifierBit modifierBits`
- `uint8_t actionPayloadLength`
- `EmberAfRf4ceZrcActionBank actionBank`
- `EmberAfRf4ceZrcActionCode actionCode`
- `uint16_t actionVendorId`
- `const uint8_t * actionPayload`
- `uint16_t timeMs`

7.26.1 Detailed Description

This data structure contains the ZRC action record.

Definition at line 353 of file [rf4ce-zrc20-types.h](#).

7.26.2 Field Documentation

7.26.2.1 `uint8_t EmberAfRf4ceZrcActionRecord::pairingIndex`

Definition at line 354 of file [rf4ce-zrc20-types.h](#).

7.26.2.2 `EmberAfRf4ceZrcActionType EmberAfRf4ceZrcActionRecord::actionType`

Definition at line 355 of file [rf4ce-zrc20-types.h](#).

7.26.2.3 `EmberAfRf4ceZrcModifierBit EmberAfRf4ceZrcActionRecord::modifierBits`

Definition at line 356 of file [rf4ce-zrc20-types.h](#).

7.26.2.4 `uint8_t EmberAfRf4ceZrcActionRecord::actionPayloadLength`

Definition at line 357 of file [rf4ce-zrc20-types.h](#).

7.26.2.5 `EmberAfRf4ceZrcActionBank EmberAfRf4ceZrcActionRecord::actionBank`

Definition at line 358 of file [rf4ce-zrc20-types.h](#).

7.26.2.6 `EmberAfRf4ceZrcActionCode EmberAfRf4ceZrcActionRecord::actionCode`

Definition at line 359 of file [rf4ce-zrc20-types.h](#).

7.26.2.7 `uint16_t EmberAfRf4ceZrcActionRecord::actionVendorId`

Definition at line 360 of file [rf4ce-zrc20-types.h](#).

7.26.2.8 const uint8_t* EmberAfRf4ceZrcActionRecord::actionPayload

Definition at line 361 of file [rf4ce-zrc20-types.h](#).

7.26.2.9 uint16_t EmberAfRf4ceZrcActionRecord::timeMs

Definition at line 362 of file [rf4ce-zrc20-types.h](#).

The documentation for this struct was generated from the following file:

- [rf4ce-zrc20-types.h](#)

7.27 EmberAfRf4ceZrcCommandsSupported Struct Reference

```
#include <rf4ce-zrc11-types.h>
```

Data Fields

- uint8_t [contents](#) [EMBER_AF_RF4CE_ZRC_COMMANDS_SUPPORTED_SIZE]

7.27.1 Detailed Description

This data structure contains the ZRC 1.x command discovery data.

Definition at line 133 of file [rf4ce-zrc11-types.h](#).

7.27.2 Field Documentation

7.27.2.1 uint8_t EmberAfRf4ceZrcCommandsSupported::contents

This is the command discovery data.

Definition at line 135 of file [rf4ce-zrc11-types.h](#).

The documentation for this struct was generated from the following files:

- [rf4ce-zrc11-types.h](#)
- [rf4ce-zrc20-types.h](#)

7.28 EmberAfRf4ceZrcHomeAutomationAttribute Struct Reference

```
#include <rf4ce-zrc20-types.h>
```

Data Fields

- uint8_t * [contents](#)
- uint8_t [contentsLength](#)
- uint8_t [instanceId](#)
- uint8_t [attributeId](#)

7.28.1 Detailed Description

RF4CE ZRC Home Automation attribute.

Definition at line 427 of file [rf4ce-zrc20-types.h](#).

7.28.2 Field Documentation

7.28.2.1 `uint8_t* EmberAfRf4ceZrcHomeAutomationAttribute::contents`

Definition at line 428 of file [rf4ce-zrc20-types.h](#).

7.28.2.2 `uint8_t EmberAfRf4ceZrcHomeAutomationAttribute::contentsLength`

Definition at line 429 of file [rf4ce-zrc20-types.h](#).

7.28.2.3 `uint8_t EmberAfRf4ceZrcHomeAutomationAttribute::instanceId`

Definition at line 430 of file [rf4ce-zrc20-types.h](#).

7.28.2.4 `uint8_t EmberAfRf4ceZrcHomeAutomationAttribute::attributeId`

Definition at line 431 of file [rf4ce-zrc20-types.h](#).

The documentation for this struct was generated from the following file:

- [rf4ce-zrc20-types.h](#)

7.29 EmberAfRf4ceZrcHomeAutomationSupported Struct Reference

```
#include <rf4ce-zrc20-types.h>
```

Data Fields

- `uint8_t contents [32]`

7.29.1 Detailed Description

RF4CE ZRC Home Automation supported.

Definition at line 420 of file [rf4ce-zrc20-types.h](#).

7.29.2 Field Documentation

7.29.2.1 `uint8_t EmberAfRf4ceZrcHomeAutomationSupported::contents[32]`

Definition at line 421 of file [rf4ce-zrc20-types.h](#).

The documentation for this struct was generated from the following file:

- [rf4ce-zrc20-types.h](#)

7.30 EmberAfRf4ceZrcMappableAction Struct Reference

```
#include <rf4ce-zrc20-types.h>
```

Data Fields

- [EmberAfRf4ceDeviceType actionDeviceType](#)
- [EmberAfRf4ceZrcActionBank actionBank](#)
- [EmberAfRf4ceZrcActionCode actionCode](#)

7.30.1 Detailed Description

RF4CE ZRC Mappable Action.

Definition at line [368](#) of file [rf4ce-zrc20-types.h](#).

7.30.2 Field Documentation

7.30.2.1 EmberAfRf4ceDeviceType EmberAfRf4ceZrcMappableAction::actionDeviceType

Definition at line [370](#) of file [rf4ce-zrc20-types.h](#).

7.30.2.2 EmberAfRf4ceZrcActionBank EmberAfRf4ceZrcMappableAction::actionBank

Definition at line [371](#) of file [rf4ce-zrc20-types.h](#).

7.30.2.3 EmberAfRf4ceZrcActionCode EmberAfRf4ceZrcMappableAction::actionCode

Definition at line [372](#) of file [rf4ce-zrc20-types.h](#).

The documentation for this struct was generated from the following file:

- [rf4ce-zrc20-types.h](#)

7.31 EmberAfRf4ceZrcUserControlRecord Struct Reference

```
#include <rf4ce-zrc11-types.h>
```

Data Fields

- [uint8_t pairingIndex](#)
- [EmberAfRf4ceZrcCommandCode commandCode](#)
- [EmberAfRf4ceUserControlCode rcCommandCode](#)
- [const uint8_t * rcCommandPayload](#)

- `uint8_t rcCommandPayloadLength`
- `uint16_t timeMs`

7.31.1 Detailed Description

This data structure contains the ZRC 1.x user control record.

Definition at line 116 of file `rf4ce-zrc11-types.h`.

7.31.2 Field Documentation

7.31.2.1 `uint8_t EmberAfRf4ceZrcUserControlRecord::pairingIndex`

Definition at line 117 of file `rf4ce-zrc11-types.h`.

7.31.2.2 `EmberAfRf4ceZrcCommandCode EmberAfRf4ceZrcUserControlRecord::commandCode`

Definition at line 118 of file `rf4ce-zrc11-types.h`.

7.31.2.3 `EmberAfRf4ceUserControlCode EmberAfRf4ceZrcUserControlRecord::rcCommandCode`

Definition at line 119 of file `rf4ce-zrc11-types.h`.

7.31.2.4 `const uint8_t* EmberAfRf4ceZrcUserControlRecord::rcCommandPayload`

Definition at line 120 of file `rf4ce-zrc11-types.h`.

7.31.2.5 `uint8_t EmberAfRf4ceZrcUserControlRecord::rcCommandPayloadLength`

Definition at line 121 of file `rf4ce-zrc11-types.h`.

7.31.2.6 `uint16_t EmberAfRf4ceZrcUserControlRecord::timeMs`

Definition at line 122 of file `rf4ce-zrc11-types.h`.

The documentation for this struct was generated from the following file:

- `rf4ce-zrc11-types.h`

7.32 HaAttributesInfo Struct Reference

```
#include <rf4ce-zrc20-ha-server.h>
```

Data Fields

- `uint8_t id`
- `uint8_t length`

7.32.1 Detailed Description

Definition at line 199 of file [rf4ce-zrc20-ha-server.h](#).

The documentation for this struct was generated from the following file:

- [rf4ce-zrc20-ha-server.h](#)

Chapter 8

File Documentation

8.1 _AF_RF4CE.top File Reference

8.2 _AF_RF4CE.top

00001

8.3 callback.doc File Reference

Functions

- int `main` (MAIN_FUNCTION_PARAMETERS)
- uint16_t `halBulbPwmDriverFrequencyCallback` (void)
- void `halBulbPwmDriverInitCompleteCallback` (void)
- void `halBulbPwmDriverBlinkOnCallback` (void)
- void `halBulbPwmDriverBlinkOffCallback` (void)
- void `halBulbPwmDriverBlinkStartCallback` (void)
- void `halBulbPwmDriverBlinkStopCallback` (void)
- void `halButtonIsr` (uint8_t button, uint8_t state)
- void `emberDebugHandler` (EmberMessageBuffer message)
- void `ezspErrorHandler` (EzspStatus status)
- void `ezspWaitingForResponse` (void)
- void `ezspNoCallbacks` (void)
- void `ezspStackTokenChangedHandler` (uint16_t tokenAddress)
- void `ezspTimerHandler` (uint8_t timerId)
- void `ezspCounterRolloverHandler` (EmberCounterType type)
- void `ezspCustomFrameHandler` (uint8_t payloadLength, uint8_t *payload)
- void `ezspStackStatusHandler` (EmberStatus status)
- void `ezspEnergyScanResultHandler` (uint8_t channel, int8_t maxRssiValue)
- void `ezspNetworkFoundHandler` (EmberZigbeeNetwork *networkFound, uint8_t lastHopLqi, int8_t lastHopRssi)
- void `ezspScanCompleteHandler` (uint8_t channel, EmberStatus status)
- void `ezspChildJoinHandler` (uint8_t index, bool joining, EmberNodeId childId, EmberEUI64 childEui64, EmberNodeType childType)

- void **ezspRemoteSetBindingHandler** (EmberBindingTableEntry *entry, uint8_t index, EmberStatus policyDecision)
- void **ezspRemoteDeleteBindingHandler** (uint8_t index, EmberStatus policyDecision)
- void **ezspMessageSentHandler** (EmberOutgoingMessageType type, uint16_t indexOrDestination, EmberApsFrame *apsFrame, uint8_t messageTag, EmberStatus status, uint8_t messageLength, uint8_t *messageContents)
- void **ezspPollCompleteHandler** (EmberStatus status)
- void **ezspPollHandler** (EmberNodeId childId)
- void **ezspIncomingSenderEui64Handler** (EmberEUI64 senderEui64)
- void **ezspIncomingMessageHandler** (EmberIncomingMessageType type, EmberApsFrame *apsFrame, uint8_t lastHopLqi, int8_t lastHopRssi, EmberNodeId sender, uint8_t bindingIndex, uint8_t addressIndex, uint8_t messageLength, uint8_t *messageContents)
- void **ezspIncomingRouteRecordHandler** (EmberNodeId source, EmberEUI64 sourceEui, uint8_t lastHopLqi, int8_t lastHopRssi, uint8_t relayCount, uint8_t *relayList)
- void **ezspIncomingManyToOneRouteRequestHandler** (EmberNodeId source, EmberEUI64 longId, uint8_t cost)
- void **ezspIncomingRouteErrorHandler** (EmberStatus status, EmberNodeId target)
- void **ezspIdConflictHandler** (EmberNodeId id)
- void **ezspMacPassthroughMessageHandler** (EmberMacPassthroughType messageType, uint8_t lastHopLqi, int8_t lastHopRssi, uint8_t messageLength, uint8_t *messageContents)
- void **ezspMacFilterMatchMessageHandler** (uint8_t filterIndexMatch, EmberMacPassthroughType legacyPassthroughType, uint8_t lastHopLqi, int8_t lastHopRssi, uint8_t messageLength, uint8_t *messageContents)
- void **ezspRawTransmitCompleteHandler** (EmberStatus status)
- void **ezspSwitchNetworkKeyHandler** (uint8_t sequenceNumber)
- void **ezspZigbeeKeyEstablishmentHandler** (EmberEUI64 partner, EmberKeyStatus status)
- void **ezspTrustCenterJoinHandler** (EmberNodeId newNodeId, EmberEUI64 newNodeEui64, EmberDeviceUpdate status, EmberJoinDecision policyDecision, EmberNodeId parentOfNewNodeId)
- void **ezspGenerateCbkeKeysHandler** (EmberStatus status, EmberPublicKeyData *ephemeralPublicKey)
- void **ezspCalculateSmacsHandler** (EmberStatus status, EmberSmacData *initiatorSmac, EmberSmacData *responderSmac)
- void **ezspGenerateCbkeKeysHandler283k1** (EmberStatus status, EmberPublicKey283k1Data *ephemeralPublicKey)
- void **ezspCalculateSmacsHandler283k1** (EmberStatus status, EmberSmacData *initiatorSmac, EmberSmacData *responderSmac)
- void **ezspDsaSignHandler** (EmberStatus status, uint8_t messageLength, uint8_t *messageContents)
- void **ezspDsaVerifyHandler** (EmberStatus status)
- void **ezspMfglibRxHandler** (uint8_t linkQuality, int8_t rssi, uint8_t packetLength, uint8_t *packetContents)
- void **ezspIncomingBootloadMessageHandler** (EmberEUI64 longId, uint8_t lastHopLqi, int8_t lastHopRssi, uint8_t messageLength, uint8_t *messageContents)
- void **ezspBootloadTransmitCompleteHandler** (EmberStatus status, uint8_t messageLength, uint8_t *messageContents)
- void **ezspZllNetworkFoundHandler** (EmberZllNetwork *networkInfo, bool isDeviceInfoNull, EmberZllDeviceInfoRecord *deviceInfo, uint8_t lastHopLqi, int8_t lastHopRssi)
- void **ezspZllScanCompleteHandler** (EmberStatus status)
- void **ezspZllAddressAssignmentHandler** (EmberZllAddressAssignment *addressInfo, uint8_t lastHopLqi, int8_t lastHopRssi)
- void **ezspZllTouchLinkTargetHandler** (EmberZllNetwork *networkInfo)
- void **ezspRf4ceIncomingMessageHandler** (uint8_t pairingIndex, uint8_t profileId, uint16_t vendorId, EmberRf4ceTxOption txOptions, uint8_t messageLength, uint8_t *message)

- void `ezspRf4ceMessageSentHandler` (EmberStatus status, uint8_t pairingIndex, EmberRf4ceTxOption txOptions, uint8_t profileId, uint16_t vendorId, uint8_t messageTag, uint8_t messageLength, uint8_t *message)
- void `ezspRf4ceDiscoveryCompleteHandler` (EmberStatus status)
- void `ezspRf4ceDiscoveryRequestHandler` (EmberEUI64 ieeeAddr, uint8_t nodeCapabilities, EmberRf4ceVendorInfo *vendorInfo, EmberRf4ceApplicationInfo *appInfo, uint8_t searchDevType, uint8_t rxLinkQuality)
- void `ezspRf4ceDiscoveryResponseHandler` (bool atCapacity, uint8_t channel, EmberPanId panId, EmberEUI64 ieeeAddr, uint8_t nodeCapabilities, EmberRf4ceVendorInfo *vendorInfo, EmberRf4ceApplicationInfo *appInfo, uint8_t rxLinkQuality, uint8_t discRequestLqi)
- void `ezspRf4ceAutoDiscoveryResponseCompleteHandler` (EmberStatus status, EmberEUI64 srcIeeeAddr, uint8_t nodeCapabilities, EmberRf4ceVendorInfo *vendorInfo, EmberRf4ceApplicationInfo *appInfo, uint8_t searchDevType)
- void `ezspRf4cePairCompleteHandler` (EmberStatus status, uint8_t pairingIndex, EmberRf4ceVendorInfo *vendorInfo, EmberRf4ceApplicationInfo *appInfo)
- void `ezspRf4cePairRequestHandler` (EmberStatus status, uint8_t pairingIndex, EmberEUI64 srcIeeeAddr, uint8_t nodeCapabilities, EmberRf4ceVendorInfo *vendorInfo, EmberRf4ceApplicationInfo *appInfo, uint8_t keyExchangeTransferCount)
- void `ezspRf4ceUnpairHandler` (uint8_t pairingIndex)
- void `ezspRf4ceUnpairCompleteHandler` (uint8_t pairingIndex)
- void `ezspDGpSentHandler` (EmberStatus status, uint8_t gpepHandle)
- void `ezspGpepIncomingMessageHandler` (EmberStatus status, uint8_t gpdLink, uint8_t sequenceNumber, EmberGpAddress *addr, EmberGpSecurityLevel gpdfSecurityLevel, EmberGpKeyType gpdfSecurityKeyType, bool autoCommissioning, bool rxAfterTx, uint32_t gpdSecurityFrameCounter, uint8_t gpdCommandId, uint32_t mic, EmberGpSinkListEntry *sinkList, uint8_t gpdCommandPayloadLength, uint8_t *gpdCommandPayload)
- void `halRadioPowerUpHandler` (void)
- void `halRadioPowerDownHandler` (void)
- bool `emberAfPluginIdleSleepOkToSleepCallback` (uint32_t durationMs)
- void `emberAfPluginIdleSleepWakeUpCallback` (uint32_t durationMs)
- bool `emberAfPluginIdleSleepOkToIdleCallback` (void)
- void `emberAfPluginIdleSleepActiveCallback` (void)
- bool `emberAfPluginLowVoltageShutdownOkToShutdownCallback` (uint16_t shutdownVoltage)
- void `emberAfPluginLowVoltageShutdownPreShutdownCallback` (uint16_t shutdownVoltage)
- void `emberAfMainInitCallback` (void)
- void `emberAfMainTickCallback` (void)
- void `emberAfStackStatusCallback` (EmberStatus status)
- void `emberAfNcpInitCallback` (bool memoryAllocation)
- void `halMicrophoneCodecMsadpcmDataReadyCallback` (uint8_t *data, uint8_t length)
- void `halMicrophoneImaadpcmDataReadyCallback` (uint8_t *data, uint8_t length)
- bool `emberAfPluginRf4ceGdpZrc20StartConfigurationCallback` (bool isOriginator, uint8_t pairingIndex)
- void `emberAfPluginRf4ceGdpZrc20BindingCompleteCallback` (EmberAfRf4ceGdpBindingStatus status, uint8_t pairingIndex)
- void `emberAfPluginRf4ceGdpStartValidationCallback` (uint8_t pairingIndex)
- void `emberAfPluginRf4ceGdpBindingCompleteCallback` (EmberAfRf4ceGdpBindingStatus status, uint8_t pairingIndex)
- bool `emberAfPluginRf4ceGdpIncomingBindProxyCallback` (const EmberEUI64 ieeeAddr)
- void `emberAfPluginRf4ceGdpHeartbeatPollingEstablishedCallback` (uint8_t pairingIndex, EmberAfRf4ceGdpPollingTrigger triggers)
- void `emberAfPluginRf4ceGdpIdentifyCallback` (EmberAfRf4ceGdpClientNotificationIdentifyFlags flags, uint16_t timeS)

- void `emberAfPluginRf4ceGdpIdentifyClientFoundCallback` (`EmberAfRf4ceGdpClientNotification-IdentifyFlags` flags)
- void `emberAfPluginRf4ceGdpKeyExchangeCompleteCallback` (`EmberStatus` status)
- bool `emberAfPluginRf4ceGdpVendorSpecificKeyExchangeCallback` (`uint8_t` initiatorVendorSpecificParam, `uint8_t` *responderVendorSpecificParam, `uint8_t` *sharedSecret)
- void `emberAfPluginRf4ceMsoStartValidationCallback` (`uint8_t` pairingIndex)
- void `emberAfPluginRf4ceMsoBindingCompleteCallback` (`EmberAfRf4ceMsoBindingStatus` status, `uint8_t` pairingIndex)
- void `emberAfPluginRf4ceMsoUserControlCallback` (const `EmberAfRf4ceMsoUserControlRecord` *record)
- `EmberAfRf4ceStatus` `emberAfPluginRf4ceMsoGetIrRfDatabaseAttributeCallback` (`uint8_t` pairingIndex, `uint8_t` entryIndex, `uint8_t` *valueLength, `uint8_t` *value)
- bool `emberAfPluginRf4ceMsoHaveIrRfDatabaseAttributeCallback` (`uint8_t` pairingIndex, `uint8_t` entryIndex)
- void `emberAfPluginRf4ceMsoIncomingIrRfDatabaseAttributeCallback` (`uint8_t` pairingIndex, `uint8_t` entryIndex, `uint8_t` valueLength, const `uint8_t` *value)
- `EmberStatus` `emberAfPluginRf4ceMsoGetIrRfDatabaseEntryCallback` (`EmberAfRf4ceMsoKeyCode` keyCode, `EmberAfRf4ceMsoIrRfDatabaseEntry` *entry)
- bool `emberAfPluginRf4ceProfileGdpMessageSentCallback` (`uint8_t` pairingIndex, `uint8_t` profileId, `uint16_t` vendorId, `uint8_t` messageTag, const `uint8_t` *message, `uint8_t` messageLength, `EmberStatus` status)
- void `emberAfPluginRf4ceProfileRemoteControl11MessageSentCallback` (`uint8_t` pairingIndex, `uint16_t` vendorId, `uint8_t` messageTag, const `uint8_t` *message, `uint8_t` messageLength, `EmberStatus` status)
- void `emberAfPluginRf4ceProfileZrc20MessageSentCallback` (`uint8_t` pairingIndex, `uint16_t` vendorId, `uint8_t` messageTag, const `uint8_t` *message, `uint8_t` messageLength, `EmberStatus` status)
- void `emberAfPluginRf4ceProfileMsoMessageSentCallback` (`uint8_t` pairingIndex, `uint16_t` vendorId, `uint8_t` messageTag, const `uint8_t` *message, `uint8_t` messageLength, `EmberStatus` status)
- void `emberAfPluginRf4ceProfileMessageSentCallback` (`uint8_t` pairingIndex, `uint8_t` profileId, `uint16_t` vendorId, `uint8_t` messageTag, const `uint8_t` *message, `uint8_t` messageLength, `EmberStatus` status)
- bool `emberAfPluginRf4ceProfileGdpIncomingMessageCallback` (`uint8_t` pairingIndex, `uint8_t` profileId, `uint16_t` vendorId, `EmberRf4ceTxOption` txOptions, const `uint8_t` *message, `uint8_t` messageLength)
- void `emberAfPluginRf4ceProfileRemoteControl11IncomingMessageCallback` (`uint8_t` pairingIndex, `uint16_t` vendorId, `EmberRf4ceTxOption` txOptions, const `uint8_t` *message, `uint8_t` messageLength)
- void `emberAfPluginRf4ceProfileZrc20IncomingMessageCallback` (`uint8_t` pairingIndex, `uint16_t` vendorId, `EmberRf4ceTxOption` txOptions, const `uint8_t` *message, `uint8_t` messageLength)
- void `emberAfPluginRf4ceProfileMsoIncomingMessageCallback` (`uint8_t` pairingIndex, `uint16_t` vendorId, `EmberRf4ceTxOption` txOptions, const `uint8_t` *message, `uint8_t` messageLength)
- void `emberAfPluginRf4ceProfileIncomingMessageCallback` (`uint8_t` pairingIndex, `uint8_t` profileId, `uint16_t` vendorId, `EmberRf4ceTxOption` txOptions, const `uint8_t` *message, `uint8_t` messageLength)
- bool `emberAfPluginRf4ceProfileGdpDiscoveryRequestCallback` (const `EmberEUI64` ieeeAddr, `uint8_t` nodeCapabilities, const `EmberRf4ceVendorInfo` *vendorInfo, const `EmberRf4ceApplicationInfo` *appInfo, `uint8_t` searchDevType, `uint8_t` rxLinkQuality)
- bool `emberAfPluginRf4ceProfileRemoteControl11DiscoveryRequestCallback` (const `EmberEUI64` ieeeAddr, `uint8_t` nodeCapabilities, const `EmberRf4ceVendorInfo` *vendorInfo, const `EmberRf4ceApplicationInfo` *appInfo, `uint8_t` searchDevType, `uint8_t` rxLinkQuality)
- bool `emberAfPluginRf4ceProfileZrc20DiscoveryRequestCallback` (const `EmberEUI64` ieeeAddr, `uint8_t` nodeCapabilities, const `EmberRf4ceVendorInfo` *vendorInfo, const `EmberRf4ceApplicationInfo` *appInfo, `uint8_t` searchDevType, `uint8_t` rxLinkQuality)

- `bool emberAfPluginRf4ceProfileMsoDiscoveryRequestCallback (const EmberEUI64 ieeeAddr, uint8_t nodeCapabilities, const EmberRf4ceVendorInfo *vendorInfo, const EmberRf4ceApplicationInfo *appInfo, uint8_t searchDevType, uint8_t rxLinkQuality)`
- `bool emberAfPluginRf4ceProfileGdpDiscoveryResponseCallback (bool atCapacity, uint8_t channel, EmberPanId panId, const EmberEUI64 ieeeAddr, uint8_t nodeCapabilities, const EmberRf4ceVendorInfo *vendorInfo, const EmberRf4ceApplicationInfo *appInfo, uint8_t rxLinkQuality, uint8_t discRequestLqi)`
- `bool emberAfPluginRf4ceProfileRemoteControl11DiscoveryResponseCallback (bool atCapacity, uint8_t channel, EmberPanId panId, const EmberEUI64 ieeeAddr, uint8_t nodeCapabilities, const EmberRf4ceVendorInfo *vendorInfo, const EmberRf4ceApplicationInfo *appInfo, uint8_t rxLinkQuality, uint8_t discRequestLqi)`
- `bool emberAfPluginRf4ceProfileZrc20DiscoveryResponseCallback (bool atCapacity, uint8_t channel, EmberPanId panId, const EmberEUI64 ieeeAddr, uint8_t nodeCapabilities, const EmberRf4ceVendorInfo *vendorInfo, const EmberRf4ceApplicationInfo *appInfo, uint8_t rxLinkQuality, uint8_t discRequestLqi)`
- `bool emberAfPluginRf4ceProfileMsoDiscoveryResponseCallback (bool atCapacity, uint8_t channel, EmberPanId panId, const EmberEUI64 ieeeAddr, uint8_t nodeCapabilities, const EmberRf4ceVendorInfo *vendorInfo, const EmberRf4ceApplicationInfo *appInfo, uint8_t rxLinkQuality, uint8_t discRequestLqi)`
- `void emberAfPluginRf4ceProfileGdpDiscoveryCompleteCallback (EmberStatus status)`
- `void emberAfPluginRf4ceProfileRemoteControl11DiscoveryCompleteCallback (EmberStatus status)`
- `void emberAfPluginRf4ceProfileZrc20DiscoveryCompleteCallback (EmberStatus status)`
- `void emberAfPluginRf4ceProfileMsoDiscoveryCompleteCallback (EmberStatus status)`
- `void emberAfPluginRf4ceProfileGdpAutoDiscoveryResponseCompleteCallback (EmberStatus status, const EmberEUI64 srcIeeeAddr, uint8_t nodeCapabilities, const EmberRf4ceVendorInfo *vendorInfo, const EmberRf4ceApplicationInfo *appInfo, uint8_t searchDevType)`
- `void emberAfPluginRf4ceProfileRemoteControl11AutoDiscoveryResponseCompleteCallback (EmberStatus status, const EmberEUI64 srcIeeeAddr, uint8_t nodeCapabilities, const EmberRf4ceVendorInfo *vendorInfo, const EmberRf4ceApplicationInfo *appInfo, uint8_t searchDevType)`
- `void emberAfPluginRf4ceProfileZrc20AutoDiscoveryResponseCompleteCallback (EmberStatus status, const EmberEUI64 srcIeeeAddr, uint8_t nodeCapabilities, const EmberRf4ceVendorInfo *vendorInfo, const EmberRf4ceApplicationInfo *appInfo, uint8_t searchDevType)`
- `void emberAfPluginRf4ceProfileMsoAutoDiscoveryResponseCompleteCallback (EmberStatus status, const EmberEUI64 srcIeeeAddr, uint8_t nodeCapabilities, const EmberRf4ceVendorInfo *vendorInfo, const EmberRf4ceApplicationInfo *appInfo, uint8_t searchDevType)`
- `bool emberAfPluginRf4ceProfileGdpPairRequestCallback (EmberStatus status, uint8_t pairingIndex, const EmberEUI64 sourceIeeeAddr, uint8_t nodeCapabilities, const EmberRf4ceVendorInfo *vendorInfo, const EmberRf4ceApplicationInfo *appInfo, uint8_t keyExchangeTransferCount)`
- `bool emberAfPluginRf4ceProfileRemoteControl11PairRequestCallback (EmberStatus status, uint8_t pairingIndex, const EmberEUI64 sourceIeeeAddr, uint8_t nodeCapabilities, const EmberRf4ceVendorInfo *vendorInfo, const EmberRf4ceApplicationInfo *appInfo, uint8_t keyExchangeTransferCount)`
- `bool emberAfPluginRf4ceProfileZrc20PairRequestCallback (EmberStatus status, uint8_t pairingIndex, const EmberEUI64 sourceIeeeAddr, uint8_t nodeCapabilities, const EmberRf4ceVendorInfo *vendorInfo, const EmberRf4ceApplicationInfo *appInfo, uint8_t keyExchangeTransferCount)`
- `bool emberAfPluginRf4ceProfileMsoPairRequestCallback (EmberStatus status, uint8_t pairingIndex, const EmberEUI64 sourceIeeeAddr, uint8_t nodeCapabilities, const EmberRf4ceVendorInfo *vendorInfo, const EmberRf4ceApplicationInfo *appInfo, uint8_t keyExchangeTransferCount)`
- `void emberAfPluginRf4ceProfileGdpPairCompleteCallback (EmberStatus status, uint8_t pairingIndex, const EmberRf4ceVendorInfo *vendorInfo, const EmberRf4ceApplicationInfo *appInfo)`
- `void emberAfPluginRf4ceProfileRemoteControl11PairCompleteCallback (EmberStatus status, uint8_t pairingIndex, const EmberRf4ceVendorInfo *vendorInfo, const EmberRf4ceApplicationInfo *appInfo)`

- void `emberAfPluginRf4ceProfileZrc20PairCompleteCallback` (EmberStatus status, uint8_t pairingIndex, const EmberRf4ceVendorInfo *vendorInfo, const EmberRf4ceApplicationInfo *appInfo)
- void `emberAfPluginRf4ceProfileMsoPairCompleteCallback` (EmberStatus status, uint8_t pairingIndex, const EmberRf4ceVendorInfo *vendorInfo, const EmberRf4ceApplicationInfo *appInfo)
- void `emberRf4ceMessageSentHandler` (EmberStatus status, uint8_t pairingIndex, uint8_t txOptions, uint8_t profileId, uint16_t vendorId, uint8_t messageTag, uint8_t messageLength, uint8_t *message)
- void `emberRf4ceIncomingMessageHandler` (uint8_t pairingIndex, uint8_t profileId, uint16_t vendorId, EmberRf4ceTxOption txOptions, uint8_t messageLength, uint8_t *message)
- void `emberRf4ceDiscoveryCompleteHandler` (EmberStatus status)
- bool `emberRf4ceDiscoveryRequestHandler` (EmberEUI64 srcIeeeAddr, uint8_t nodeCapabilities, EmberRf4ceVendorInfo *vendorInfo, EmberRf4ceApplicationInfo *appInfo, uint8_t searchDevType, uint8_t rxLinkQuality)
- bool `emberRf4ceDiscoveryResponseHandler` (bool atCapacity, uint8_t channel, EmberPanId panId, EmberEUI64 srcIeeeAddr, uint8_t nodeCapabilities, EmberRf4ceVendorInfo *vendorInfo, EmberRf4ceApplicationInfo *appInfo, uint8_t rxLinkQuality, uint8_t discRequestLqi)
- void `emberRf4ceAutoDiscoveryResponseCompleteHandler` (EmberStatus status, EmberEUI64 srcIeeeAddr, uint8_t nodeCapabilities, EmberRf4ceVendorInfo *vendorInfo, EmberRf4ceApplicationInfo *appInfo, uint8_t searchDevType)
- void `emberRf4cePairCompleteHandler` (EmberStatus status, uint8_t pairingIndex, EmberRf4ceVendorInfo *vendorInfo, EmberRf4ceApplicationInfo *appInfo)
- bool `emberRf4cePairRequestHandler` (EmberStatus status, uint8_t pairingIndex, EmberEUI64 srcIeeeAddr, uint8_t nodeCapabilities, EmberRf4ceVendorInfo *vendorInfo, EmberRf4ceApplicationInfo *appInfo, uint8_t keyExchangeTransferCount)
- void `emberRf4ceUnpairHandler` (uint8_t pairingIndex)
- void `emberRf4ceUnpairCompleteHandler` (uint8_t pairingIndex)
- void `emberAfPluginRf4ceTargetCommunicationHostBindingRequestCallback` (void)
- void `emberAfPluginRf4ceZrc11PairingCompleteCallback` (EmberStatus status, uint8_t pairingIndex, const EmberEUI64 eui64, const EmberRf4ceVendorInfo *vendorInfo, const EmberRf4ceApplicationInfo *applicationInfo)
- void `emberAfPluginRf4ceZrc11UserControlCallback` (const `EmberAfRf4ceZrcUserControlRecord` *record)
- void `emberAfPluginRf4ceZrc11CommandDiscoveryResponseCallback` (EmberStatus status, const `EmberAfRf4ceZrcCommandsSupported` *commandsSupported)
- void `emberAfPluginRf4ceZrc20ActionCallback` (const `EmberAfRf4ceZrcActionRecord` *record)
- void `emberAfPluginRf4ceZrc20HaActionCallback` (const `EmberAfRf4ceZrcActionRecord` *record)
- void `emberAfPluginRf4ceZrc20LegacyCommandDiscoveryCompleteCallback` (EmberStatus status, const `EmberAfRf4ceZrcCommandsSupported` *commandsSupported)
- void `emberAfPluginRf4ceZrc20ActionMappingsNegotiationCompleteCallback` (EmberStatus status)
- void `emberAfPluginRf4ceZrc20IncomingMappableActionCallback` (uint8_t pairingIndex, uint16_t entryIndex, `EmberAfRf4ceZrcMappableAction` *mappableAction)
- EmberStatus `emberAfPluginRf4ceZrc20GetMappableActionCallback` (uint8_t pairingIndex, uint16_t entryIndex, `EmberAfRf4ceZrcMappableAction` *mappableAction)
- uint16_t `emberAfPluginRf4ceZrc20GetMappableActionCountCallback` (uint8_t pairingIndex)
- void `emberAfPluginRf4ceZrc20IncomingActionMappingCallback` (uint8_t pairingIndex, uint16_t entryIndex, `EmberAfRf4ceZrcActionMapping` *actionMapping)
- EmberStatus `emberAfPluginRf4ceZrc20GetActionMappingCallback` (uint8_t pairingIndex, uint16_t entryIndex, `EmberAfRf4ceZrcActionMapping` *actionMapping)
- EmberStatus `emberAfPluginRf4ceZrc20SetActionMappingCallback` (uint8_t pairingIndex, uint16_t entryIndex, `EmberAfRf4ceZrcActionMapping` *actionMapping)
- void `emberAfPluginRf4ceZrc20HomeAutomationSupportedAnnouncementCompleteCallback` (EmberStatus status)

- void `emberAfPluginRf4ceZrc20IncomingHomeAutomationSupportedCallback` (uint8_t pairingIndex, uint8_t haInstanceId, `EmberAfRf4ceZrcHomeAutomationSupported` *haSupported)
- EmberStatus `emberAfPluginRf4ceZrc20GetHomeAutomationSupportedCallback` (uint8_t pairingIndex, uint8_t haInstanceId, `EmberAfRf4ceZrcHomeAutomationSupported` *haSupported)
- `EmberAfRf4ceGdpAttributeStatus` `emberAfPluginRf4ceZrc20GetHomeAutomationAttributeCallback` (uint8_t pairingIndex, uint8_t haInstanceId, uint8_t haAttributeId, `EmberAfRf4ceZrcHomeAutomationAttribute` *haAttribute)
- void `emberAfPluginRf4ceZrc20PullHomeAutomationAttributeCompleteCallback` (`EmberAfRf4ceGdpAttributeStatus` responseStatus, `EmberAfRf4ceZrcHomeAutomationAttribute` *haAttribute)
- void `emberCounterHandler` (EmberCounterType type, uint8_t data)
- void `emberStackStatusHandler` (EmberStatus status)
- void `emberEnergyScanResultHandler` (uint8_t channel, int8_t maxRssiValue)
- void `emberRadioNeedsCalibratingHandler` (void)
- void `halNcpIsAwakeIsr` (bool isAwake)
- void `halSimEepromCallback` (EmberStatus status)

8.4 lv-shutdown.h File Reference

Functions

- uint16_t `emberAfPluginLowVoltageShutdownGetVoltage` (void)
- void `emberAfPluginLowVoltageShutdownEnable` (bool enable)
- bool `emberAfPluginLowVoltageShutdownEnabled` (void)
- void `emberAfPluginLowVoltageShutdownForceShutdown` (void)

8.4.1 Function Documentation

8.4.1.1 uint16_t `emberAfPluginLowVoltageShutdownGetVoltage` (void)

8.4.1.2 void `emberAfPluginLowVoltageShutdownEnable` (bool enable)

8.4.1.3 bool `emberAfPluginLowVoltageShutdownEnabled` (void)

8.4.1.4 void `emberAfPluginLowVoltageShutdownForceShutdown` (void)

8.5 lv-shutdown.h

```

00001 // Copyright 2014 Silicon Laboratories, Inc.
00002
00003 // Obtain a single reading of the VDD voltage (used to check for shutdown)
00004 // Result is in millivolts.
00005 uint16_t emberAfPluginLowVoltageShutdownGetVoltage
    (void);
00006
00007 // Enable or disable shutdown logic at runtime for testing purposes
00008 void emberAfPluginLowVoltageShutdownEnable
    (bool enable);
00009
00010 // Check whether the shutdown logic has is enabled at runtime (as it can be
00011 // artificially disabled)
00012 bool emberAfPluginLowVoltageShutdownEnabled
    (void);
00013
00014 // Force a shutdown event (regardless of VDD) for testing
00015 void emberAfPluginLowVoltageShutdownForceShutdown
    (void);

```

8.6 rf4ce-gdp-attributes.h File Reference

Data Structures

- struct [EmAfRf4ceGdpAttributeDescriptor](#)
- struct [EmAfRf4ceGdpAttributes](#)

Macros

- #define [MAX_COMMAND_ATTRIBUTES](#)
- #define [APL_GDP_VERSION_DEFAULT](#)
- #define [APL_GDP_KEY_EXCHANGE_TRANSFER_COUNT_DEFAULT](#)
- #define [APL_GDP_POWER_STATUS_DEFAULT](#)
- #define [APL_GDP_MAX_PAIRING_CANDIDATES_DEFAULT](#)
- #define [APL_GDP_AUTO_CHECK_VALIDATION_PERIOD_DEFAULT](#)
- #define [APL_GDP_LINK_LOST_WAIT_TIME_DEFAULT](#)
- #define [APL_GDP_IDENTIFICATION_CAPABILITIES_DEFAULT](#)
- #define [APL_GDP_BINDING_RECIPIENT_VALIDATION_WAIT_TIME_DEFAULT](#)
- #define [APL_GDP_BINDING_ORIGINATOR_VALIDATION_WAIT_TIME_DEFAULT](#)
- #define [APL_GDP_VERSION_SIZE](#)
- #define [APL_GDP_CAPABILITIES_SIZE](#)
- #define [APL_GDP_KEY_EXCHANGE_TRANSFER_COUNT_SIZE](#)
- #define [APL_GDP_POWER_STATUS_SIZE](#)
- #define [APL_GDP_POLL_CONSTRAINTS_SIZE](#)
- #define [APL_GDP_POLL_CONFIGURATION_SIZE](#)
- #define [APL_GDP_MAX_PAIRING_CANDIDATES_SIZE](#)
- #define [APL_GDP_AUTO_CHECK_VALIDATION_PERIOD_SIZE](#)
- #define [APL_GDP_BINDING_RECIPIENT_VALIDATION_WAIT_TIME_SIZE](#)
- #define [APL_GDP_BINDING_ORIGINATOR_VALIDATION_WAIT_TIME_SIZE](#)
- #define [APL_GDP_LINK_LOST_WAIT_TIME_SIZE](#)
- #define [APL_GDP_IDENTIFICATION_CAPABILITIES_SIZE](#)
- #define [MAX_GDP_ATTRIBUTE_SIZE](#)
- #define [GDP_ATTRIBUTES_COUNT](#)
- #define [MIN_GDP_ATTRIBUTE_ID](#)
- #define [MAX_GDP_ATTRIBUTE_ID](#)
- #define [POLL_CONSTRAINT_RECORD_NUMBER_OFFSET](#)
- #define [POLL_CONSTRAINT_RECORD_NUMBER_LENGTH](#)
- #define [POLL_CONSTRAINT_RECORD_OFFSET](#)
- #define [POLL_CONSTRAINT_RECORD_LENGTH](#)
- #define [POLL_CONSTRAINT_RECORD_POLLING_METHOD_ID_OFFSET](#)
- #define [POLL_CONSTRAINT_RECORD_POLLING_METHOD_ID_LENGTH](#)
- #define [POLL_CONSTRAINT_RECORD_POLLING_TRIGGER_CAPABILITIES_OFFSET](#)
- #define [POLL_CONSTRAINT_RECORD_POLLING_TRIGGER_CAPABILITIES_LENGTH](#)
- #define [POLL_CONSTRAINT_RECORD_MIN_POLLING_KEY_PRESS_COUNT_OFFSET](#)
- #define [POLL_CONSTRAINT_RECORD_MIN_POLLING_KEY_PRESS_COUNT_LENGTH](#)
- #define [POLL_CONSTRAINT_RECORD_MAX_POLLING_KEY_PRESS_COUNT_OFFSET](#)
- #define [POLL_CONSTRAINT_RECORD_MAX_POLLING_KEY_PRESS_COUNT_LENGTH](#)
- #define [POLL_CONSTRAINT_RECORD_MIN_POLLING_TIME_INTERVAL_OFFSET](#)
- #define [POLL_CONSTRAINT_RECORD_MIN_POLLING_TIME_INTERVAL_LENGTH](#)
- #define [POLL_CONSTRAINT_RECORD_MAX_POLLING_TIME_INTERVAL_OFFSET](#)

- #define POLL_CONSTRAINT_RECORD_MAX_POLLING_TIME_INTERVAL_LENGTH
- #define POLL_CONSTRAINT_RECORD_MIN_POLLING_TIME_INTERVAL_LOWER_BOUND
- #define POLL_CONSTRAINT_RECORD_MIN_POLLING_TIME_INTERVAL_UPPER_BOUND
- #define POLL_CONSTRAINT_RECORD_MAX_POLLING_TIME_INTERVAL_LOWER_BOUND
- #define POLL_CONSTRAINT_RECORD_MAX_POLLING_TIME_INTERVAL_UPPER_BOUND
- #define POLL_CONFIGURATION_POLLING_METHOD_ID_OFFSET
- #define POLL_CONFIGURATION_POLLING_METHOD_ID_LENGTH
- #define POLL_CONFIGURATION_POLLING_TRIGGER_CONFIG_OFFSET
- #define POLL_CONFIGURATION_POLLING_TRIGGER_CONFIG_LENGTH
- #define POLL_CONFIGURATION_POLLING_KEY_PRESS_COUNTER_OFFSET
- #define POLL_CONFIGURATION_POLLING_KEY_PRESS_COUNTER_LENGTH
- #define POLL_CONFIGURATION_POLLING_TIME_INTERVAL_OFFSET
- #define POLL_CONFIGURATION_POLLING_TIME_INTERVAL_LENGTH
- #define POLL_CONFIGURATION_POLLING_TIMEOUT_OFFSET
- #define POLL_CONFIGURATION_POLLING_TIMEOUT_LENGTH
- #define MIN_POLLING_KEY_PRESS_COUNTER
- #define MAX_POLLING_KEY_PRESS_COUNTER
- #define SUPPORTED_TRIGGERS
- #define APL_POLL_CONSTRAINTS_DEFAULT
- #define IDENTIFICATION_CAPABILITIES_RESERVED_MASK
- #define IDENTIFICATION_CAPABILITIES_SUPPORT_FLASH_LIGHT_BIT
- #define IDENTIFICATION_CAPABILITIES_SUPPORT_MAKE_SOUND_BIT
- #define IDENTIFICATION_CAPABILITIES_SUPPORT_VIBRATE_BIT
- #define ATTRIBUTE_HAS_REMOTE_GET_ACCESS_BIT
- #define ATTRIBUTE_HAS_REMOTE_SET_ACCESS_BIT
- #define ATTRIBUTE_HAS_REMOTE_PUSH_ACCESS_BIT
- #define ATTRIBUTE_HAS_REMOTE_PULL_ACCESS_BIT
- #define ATTRIBUTE_IS_TWO_DIMENSIONAL_ARRAYED
- #define IS_ARRAY_ATTRIBUTE(attributeId)
- #define GDP_ATTRIBUTE_ONE_DIMENSIONAL_ARRAY_TEST_SIZE
- #define GDP_ATTRIBUTE_ONE_DIMENSIONAL_ARRAY_TEST_DIMENSION
- #define GDP_ATTRIBUTE_TWO_DIMENSIONAL_ARRAY_TEST_SIZE
- #define GDP_ATTRIBUTE_TWO_DIMENSIONAL_ARRAY_TEST_FIRST_DIMENSION
- #define GDP_ATTRIBUTE_TWO_DIMENSIONAL_ARRAY_TEST_SECOND_DIMENSION
- #define WRITE_ACCESS
- #define READ_ACCESS
- #define emAfRf4ceGdpSetLocalAttribute(attrId, entryId, val)
- #define emAfRf4ceGdpGetLocalAttribute(attrId, entryId, val)
- #define emAfRf4ceGdpSetRemoteAttribute(attrId, entryId, val)
- #define emAfRf4ceGdpGetRemoteAttribute(attrId, entryId, val)

Functions

- void emAfRf4ceGdpClearRemoteAttributes (void)
- void emAfRf4ceGdpGetOrSetAttribute (EmAfRf4ceGdpAttributes *attributes, uint8_t attrId, uint16_t entryId, bool isGet, uint8_t *val)

Variables

- EmAfRf4ceGdpAttributes emAfRf4ceGdpLocalNodeAttributes
- EmAfRf4ceGdpAttributes emAfRf4ceGdpRemoteNodeAttributes

8.6.1 Macro Definition Documentation

8.6.1.1 #define MAX_COMMAND_ATTRIBUTES

Definition at line 9 of file [rf4ce-gdp-attributes.h](#).

8.6.1.2 #define APL_GDP_VERSION_DEFAULT

Definition at line 12 of file [rf4ce-gdp-attributes.h](#).

8.6.1.3 #define APL_GDP_KEY_EXCHANGE_TRANSFER_COUNT_DEFAULT

Definition at line 13 of file [rf4ce-gdp-attributes.h](#).

8.6.1.4 #define APL_GDP_POWER_STATUS_DEFAULT

Definition at line 14 of file [rf4ce-gdp-attributes.h](#).

8.6.1.5 #define APL_GDP_MAX_PAIRING_CANDIDATES_DEFAULT

Definition at line 15 of file [rf4ce-gdp-attributes.h](#).

8.6.1.6 #define APL_GDP_AUTO_CHECK_VALIDATION_PERIOD_DEFAULT

Definition at line 16 of file [rf4ce-gdp-attributes.h](#).

8.6.1.7 #define APL_GDP_LINK_LOST_WAIT_TIME_DEFAULT

Definition at line 17 of file [rf4ce-gdp-attributes.h](#).

8.6.1.8 #define APL_GDP_IDENTIFICATION_CAPABILITIES_DEFAULT

Definition at line 18 of file [rf4ce-gdp-attributes.h](#).

8.6.1.9 #define APL_GDP_BINDING_RECIPIENT_VALIDATION_WAIT_TIME_DEFAULT

Definition at line 19 of file [rf4ce-gdp-attributes.h](#).

8.6.1.10 #define APL_GDP_BINDING_ORIGINATOR_VALIDATION_WAIT_TIME_DEFAULT

Definition at line 30 of file [rf4ce-gdp-attributes.h](#).

8.6.1.11 #define APL_GDP_VERSION_SIZE

Definition at line 34 of file [rf4ce-gdp-attributes.h](#).

8.6.1.12 #define APL_GDP_CAPABILITIES_SIZE

Definition at line 35 of file [rf4ce-gdp-attributes.h](#).

8.6.1.13 #define APL_GDP_KEY_EXCHANGE_TRANSFER_COUNT_SIZE

Definition at line 36 of file [rf4ce-gdp-attributes.h](#).

8.6.1.14 #define APL_GDP_POWER_STATUS_SIZE

Definition at line 37 of file [rf4ce-gdp-attributes.h](#).

8.6.1.15 #define APL_GDP_POLL_CONSTRAINTS_SIZE

Definition at line 42 of file [rf4ce-gdp-attributes.h](#).

8.6.1.16 #define APL_GDP_POLL_CONFIGURATION_SIZE

Definition at line 43 of file [rf4ce-gdp-attributes.h](#).

8.6.1.17 #define APL_GDP_MAX_PAIRING_CANDIDATES_SIZE

Definition at line 44 of file [rf4ce-gdp-attributes.h](#).

8.6.1.18 #define APL_GDP_AUTO_CHECK_VALIDATION_PERIOD_SIZE

Definition at line 45 of file [rf4ce-gdp-attributes.h](#).

8.6.1.19 #define APL_GDP_BINDING_RECIPIENT_VALIDATION_WAIT_TIME_SIZE

Definition at line 46 of file [rf4ce-gdp-attributes.h](#).

8.6.1.20 #define APL_GDP_BINDING_ORIGINATOR_VALIDATION_WAIT_TIME_SIZE

Definition at line 47 of file [rf4ce-gdp-attributes.h](#).

8.6.1.21 #define APL_GDP_LINK_LOST_WAIT_TIME_SIZE

Definition at line 48 of file [rf4ce-gdp-attributes.h](#).

8.6.1.22 #define APL_GDP_IDENTIFICATION_CAPABILITIES_SIZE

Definition at line 49 of file [rf4ce-gdp-attributes.h](#).

8.6.1.23 #define MAX_GDP_ATTRIBUTE_SIZE

Definition at line 52 of file [rf4ce-gdp-attributes.h](#).

8.6.1.24 #define GDP_ATTRIBUTES_COUNT

Definition at line 57 of file [rf4ce-gdp-attributes.h](#).

8.6.1.25 #define MIN_GDP_ATTRIBUTE_ID

Definition at line 60 of file [rf4ce-gdp-attributes.h](#).

8.6.1.26 #define MAX_GDP_ATTRIBUTE_ID

Definition at line 61 of file [rf4ce-gdp-attributes.h](#).

8.6.1.27 #define POLL_CONSTRAINT_RECORD_NUMBER_OFFSET

Definition at line 63 of file [rf4ce-gdp-attributes.h](#).

8.6.1.28 #define POLL_CONSTRAINT_RECORD_NUMBER_LENGTH

Definition at line 64 of file [rf4ce-gdp-attributes.h](#).

8.6.1.29 #define POLL_CONSTRAINT_RECORD_OFFSET

Definition at line 65 of file [rf4ce-gdp-attributes.h](#).

8.6.1.30 #define POLL_CONSTRAINT_RECORD_LENGTH

Definition at line 66 of file [rf4ce-gdp-attributes.h](#).

8.6.1.31 #define POLL_CONSTRAINT_RECORD_POLLING_METHOD_ID_OFFSET

Definition at line 69 of file [rf4ce-gdp-attributes.h](#).

8.6.1.32 #define POLL_CONSTRAINT_RECORD_POLLING_METHOD_ID_LENGTH

Definition at line 70 of file [rf4ce-gdp-attributes.h](#).

8.6.1.33 #define POLL_CONSTRAINT_RECORD_POLLING_TRIGGER_CAPABILITIES_OFFSET

Definition at line 71 of file [rf4ce-gdp-attributes.h](#).

8.6.1.34 #define POLL_CONSTRAINT_RECORD_POLLING_TRIGGER_CAPABILITIES_LENGTH

Definition at line 72 of file [rf4ce-gdp-attributes.h](#).

8.6.1.35 #define POLL_CONSTRAINT_RECORD_MIN_POLLING_KEY_PRESS_COUNT_OFFSET

Definition at line 73 of file [rf4ce-gdp-attributes.h](#).

8.6.1.36 #define POLL_CONSTRAINT_RECORD_MIN_POLLING_KEY_PRESS_COUNT_LENGTH

Definition at line 74 of file [rf4ce-gdp-attributes.h](#).

8.6.1.37 #define POLL_CONSTRAINT_RECORD_MAX_POLLING_KEY_PRESS_COUNT_OFFSET

Definition at line 75 of file [rf4ce-gdp-attributes.h](#).

8.6.1.38 #define POLL_CONSTRAINT_RECORD_MAX_POLLING_KEY_PRESS_COUNT_LENGTH

Definition at line 76 of file [rf4ce-gdp-attributes.h](#).

8.6.1.39 #define POLL_CONSTRAINT_RECORD_MIN_POLLING_TIME_INTERVAL_OFFSET

Definition at line 77 of file [rf4ce-gdp-attributes.h](#).

8.6.1.40 #define POLL_CONSTRAINT_RECORD_MIN_POLLING_TIME_INTERVAL_LENGTH

Definition at line 78 of file [rf4ce-gdp-attributes.h](#).

8.6.1.41 #define POLL_CONSTRAINT_RECORD_MAX_POLLING_TIME_INTERVAL_OFFSET

Definition at line 79 of file [rf4ce-gdp-attributes.h](#).

8.6.1.42 #define POLL_CONSTRAINT_RECORD_MAX_POLLING_TIME_INTERVAL_LENGTH

Definition at line 80 of file [rf4ce-gdp-attributes.h](#).

8.6.1.43 #define POLL_CONSTRAINT_RECORD_MIN_POLLING_TIME_INTERVAL_LOWER_BOUND

Definition at line 82 of file [rf4ce-gdp-attributes.h](#).

8.6.1.44 #define POLL_CONSTRAINT_RECORD_MIN_POLLING_TIME_INTERVAL_UPPER_BOUND

Definition at line 83 of file [rf4ce-gdp-attributes.h](#).

8.6.1.45 #define POLL_CONSTRAINT_RECORD_MAX_POLLING_TIME_INTERVAL_LOWER_BOUND

Definition at line 84 of file [rf4ce-gdp-attributes.h](#).

8.6.1.46 #define POLL_CONSTRAINT_RECORD_MAX_POLLING_TIME_INTERVAL_UPPER_BOUND

Definition at line 85 of file [rf4ce-gdp-attributes.h](#).

8.6.1.47 #define POLL_CONFIGURATION_POLLING_METHOD_ID_OFFSET

Definition at line 88 of file [rf4ce-gdp-attributes.h](#).

8.6.1.48 #define POLL_CONFIGURATION_POLLING_METHOD_ID_LENGTH

Definition at line 89 of file [rf4ce-gdp-attributes.h](#).

8.6.1.49 #define POLL_CONFIGURATION_POLLING_TRIGGER_CONFIG_OFFSET

Definition at line 90 of file [rf4ce-gdp-attributes.h](#).

8.6.1.50 #define POLL_CONFIGURATION_POLLING_TRIGGER_CONFIG_LENGTH

Definition at line 91 of file [rf4ce-gdp-attributes.h](#).

8.6.1.51 #define POLL_CONFIGURATION_POLLING_KEY_PRESS_COUNTER_OFFSET

Definition at line 92 of file [rf4ce-gdp-attributes.h](#).

8.6.1.52 #define POLL_CONFIGURATION_POLLING_KEY_PRESS_COUNTER_LENGTH

Definition at line 93 of file [rf4ce-gdp-attributes.h](#).

8.6.1.53 #define POLL_CONFIGURATION_POLLING_TIME_INTERVAL_OFFSET

Definition at line 94 of file [rf4ce-gdp-attributes.h](#).

8.6.1.54 #define POLL_CONFIGURATION_POLLING_TIME_INTERVAL_LENGTH

Definition at line 95 of file [rf4ce-gdp-attributes.h](#).

8.6.1.55 #define POLL_CONFIGURATION_POLLING_TIMEOUT_OFFSET

Definition at line 96 of file [rf4ce-gdp-attributes.h](#).

8.6.1.56 #define POLL_CONFIGURATION_POLLING_TIMEOUT_LENGTH

Definition at line 97 of file [rf4ce-gdp-attributes.h](#).

8.6.1.57 #define MIN_POLLING_KEY_PRESS_COUNTER

Definition at line 102 of file [rf4ce-gdp-attributes.h](#).

8.6.1.58 #define MAX_POLLING_KEY_PRESS_COUNTER

Definition at line 103 of file [rf4ce-gdp-attributes.h](#).

8.6.1.59 #define SUPPORTED_TRIGGERS

Definition at line 104 of file [rf4ce-gdp-attributes.h](#).

8.6.1.60 #define APL_POLL_CONSTRAINTS_DEFAULT

Definition at line 109 of file [rf4ce-gdp-attributes.h](#).

8.6.1.61 #define IDENTIFICATION_CAPABILITIES_RESERVED_MASK

Definition at line 125 of file [rf4ce-gdp-attributes.h](#).

8.6.1.62 #define IDENTIFICATION_CAPABILITIES_SUPPORT_FLASH_LIGHT_BIT

Definition at line 126 of file [rf4ce-gdp-attributes.h](#).

8.6.1.63 #define IDENTIFICATION_CAPABILITIES_SUPPORT_MAKE_SOUND_BIT

Definition at line 127 of file [rf4ce-gdp-attributes.h](#).

8.6.1.64 #define IDENTIFICATION_CAPABILITIES_SUPPORT_VIBRATE_BIT

Definition at line 128 of file [rf4ce-gdp-attributes.h](#).

8.6.1.65 #define ATTRIBUTE_HAS_REMOTE_GET_ACCESS_BIT

Definition at line 139 of file [rf4ce-gdp-attributes.h](#).

8.6.1.66 #define ATTRIBUTE_HAS_REMOTE_SET_ACCESS_BIT

Definition at line 140 of file [rf4ce-gdp-attributes.h](#).

8.6.1.67 #define ATTRIBUTE_HAS_REMOTE_PUSH_ACCESS_BIT

Definition at line 141 of file [rf4ce-gdp-attributes.h](#).

8.6.1.68 #define ATTRIBUTE_HAS_REMOTE_PULL_ACCESS_BIT

Definition at line 142 of file [rf4ce-gdp-attributes.h](#).

8.6.1.69 #define ATTRIBUTE_IS_TWO_DIMENSIONAL_ARRAYED

Definition at line 143 of file [rf4ce-gdp-attributes.h](#).

8.6.1.70 #define IS_ARRAY_ATTRIBUTE(*attributeld*)

Definition at line 157 of file [rf4ce-gdp-attributes.h](#).

8.6.1.71 #define GDP_ATTRIBUTE_ONE_DIMENSIONAL_ARRAY_TEST_SIZE

Definition at line 162 of file [rf4ce-gdp-attributes.h](#).

8.6.1.72 #define GDP_ATTRIBUTE_ONE_DIMENSIONAL_ARRAY_TEST_DIMENSION

Definition at line 163 of file [rf4ce-gdp-attributes.h](#).

8.6.1.73 #define GDP_ATTRIBUTE_TWO_DIMENSIONAL_ARRAY_TEST_SIZE

Definition at line 164 of file [rf4ce-gdp-attributes.h](#).

8.6.1.74 #define GDP_ATTRIBUTE_TWO_DIMENSIONAL_ARRAY_TEST_FIRST_DIMENSION

Definition at line 165 of file [rf4ce-gdp-attributes.h](#).

8.6.1.75 #define GDP_ATTRIBUTE_TWO_DIMENSIONAL_ARRAY_TEST_SECOND_DIMENSION

Definition at line 166 of file [rf4ce-gdp-attributes.h](#).

8.6.1.76 #define WRITE_ACCESS

Definition at line 190 of file [rf4ce-gdp-attributes.h](#).

8.6.1.77 #define READ_ACCESS

Definition at line 191 of file [rf4ce-gdp-attributes.h](#).

8.6.1.78 #define emAfRf4ceGdpSetLocalAttribute(attrId, entryId, val)

Definition at line 201 of file [rf4ce-gdp-attributes.h](#).

8.6.1.79 #define emAfRf4ceGdpGetLocalAttribute(attrId, entryId, val)

Definition at line 208 of file [rf4ce-gdp-attributes.h](#).

8.6.1.80 #define emAfRf4ceGdpSetRemoteAttribute(attrId, entryId, val)

Definition at line 215 of file [rf4ce-gdp-attributes.h](#).

8.6.1.81 #define emAfRf4ceGdpGetRemoteAttribute(attrId, entryId, val)

Definition at line 222 of file [rf4ce-gdp-attributes.h](#).

8.6.2 Function Documentation

8.6.2.1 void emAfRf4ceGdpClearRemoteAttributes (void)

8.6.2.2 void emAfRf4ceGdpGetOrSetAttribute (EmAfRf4ceGdpAttributes * attributes, uint8_t attrId, uint16_t entryId, bool isGet, uint8_t * val)

8.6.3 Variable Documentation

8.6.3.1 EmAfRf4ceGdpAttributes emAfRf4ceGdpLocalNodeAttributes

8.6.3.2 EmAfRf4ceGdpAttributes emAfRf4ceGdpRemoteNodeAttributes

8.7 rf4ce-gdp-attributes.h

```

00001 // Copyright 2014 Silicon Laboratories, Inc.
00002
00003
00004 #ifndef __RF4CE_GDP_ATTRIBUTES_H__
00005 #define __RF4CE_GDP_ATTRIBUTES_H__
00006
00007 // Defines the maximum number of attributes records contained in any
00008 // attribute-related command.
00009 #define MAX_COMMAND_ATTRIBUTES 5
00010
00011 // These are specified in Table 9.
00012 #define APL_GDP_VERSION_DEFAULT 0x0200
00013 #define APL_GDP_KEY_EXCHANGE_TRANSFER_COUNT_DEFAULT 3
00014 #define APL_GDP_POWER_STATUS_DEFAULT 0
00015 #define APL_GDP_MAX_PAIRING_CANDIDATES_DEFAULT 3
00016 #define APL_GDP_AUTO_CHECK_VALIDATION_PERIOD_DEFAULT 500 // msec
00017 #define APL_GDP_LINK_LOST_WAIT_TIME_DEFAULT 5000 // msec
00018 #define APL_GDP_IDENTIFICATION_CAPABILITIES_DEFAULT 0x00
00019 #define APL_GDP_BINDING_RECIPIENT_VALIDATION_WAIT_TIME_DEFAULT 15000 // msec
00020
00021 // If the Binding Originator does not support extended validation (see section

```

```

00022 // 7.2.7, this value shall be set to aplcMaxNormalValidationDuration. If the
00023 // Binding Originator supports extended validation,
00024 // aplcBindingOriginatorValidationWaitTime shall be set to
00025 // aplcMaxExtendedValidationDuration.
00026 #if defined(EMBER_AF_PLUGIN_RF4CE_GDP_EXTENDED_VALIDATION)
00027 #define APL_GDP_BINDING_ORIGINATOR_VALIDATION_WAIT_TIME_DEFAULT
00028 \
00029     APLC_MAX_EXTENDED_VALIDATION_DURATION_MS
00030 #else
00030 #define APL_GDP_BINDING_ORIGINATOR_VALIDATION_WAIT_TIME_DEFAULT
00031 \
00032     APLC_MAX_NORMAL_VALIDATION_DURATION_MS
00033 #endif
00033
00034 #define APL_GDP_VERSION_SIZE 2
00035 #define APL_GDP_CAPABILITIES_SIZE 4
00036 #define APL_GDP_KEY_EXCHANGE_TRANSFER_COUNT_SIZE 1
00037 #define APL_GDP_POWER_STATUS_SIZE 1
00038 // See section 6.2.5 (for now we have only one record since there is only one
00039 // polling method defined).
00040 // number of polling methods supports (1 byte)
00041 // + polling constraint record (13 bytes)
00042 #define APL_GDP_POLL_CONSTRAINTS_SIZE 14
00043 #define APL_GDP_POLL_CONFIGURATION_SIZE 9
00044 #define APL_GDP_MAX_PAIRING_CANDIDATES_SIZE 1
00045 #define APL_GDP_AUTO_CHECK_VALIDATION_PERIOD_SIZE 2
00046 #define APL_GDP_BINDING_RECIPIENT_VALIDATION_WAIT_TIME_SIZE 2
00047 #define APL_GDP_BINDING_ORIGINATOR_VALIDATION_WAIT_TIME_SIZE 2
00048 #define APL_GDP_LINK_LOST_WAIT_TIME_SIZE 2
00049 #define APL_GDP_IDENTIFICATION_CAPABILITIES_SIZE 1
00050
00051 // Make sure to keep these updated.
00052 #define MAX_GDP_ATTRIBUTE_SIZE 13
00053
00054 #if defined(EMBER_SCRIPTED_TEST)
00055 #define GDP_ATTRIBUTES_COUNT 12 + 4
00056 #else
00057 #define GDP_ATTRIBUTES_COUNT 12
00058 #endif
00059
00060 #define MIN_GDP_ATTRIBUTE_ID 0x80
00061 #define MAX_GDP_ATTRIBUTE_ID 0x8B
00062
00063 #define POLL_CONSTRAINT_RECORD_NUMBER_OFFSET 0
00064 #define POLL_CONSTRAINT_RECORD_NUMBER_LENGTH 1
00065 #define POLL_CONSTRAINT_RECORD_OFFSET 1
00066 #define POLL_CONSTRAINT_RECORD_LENGTH 13
00067
00068 // Poll constraint record related macros
00069 #define POLL_CONSTRAINT_RECORD_POLLING_METHOD_ID_OFFSET
00070     + POLL_CONSTRAINT_RECORD_OFFSET) (0
00070 #define POLL_CONSTRAINT_RECORD_POLLING_METHOD_ID_LENGTH 1
00071 #define POLL_CONSTRAINT_RECORD_POLLING_TRIGGER_CAPABILITIES_OFFSET
00072     + POLL_CONSTRAINT_RECORD_OFFSET) (1
00072 #define POLL_CONSTRAINT_RECORD_POLLING_TRIGGER_CAPABILITIES_LENGTH 2
00073 #define POLL_CONSTRAINT_RECORD_MIN_POLLING_KEY_PRESS_COUNT_OFFSET
00074     + POLL_CONSTRAINT_RECORD_OFFSET) (3
00074 #define POLL_CONSTRAINT_RECORD_MIN_POLLING_KEY_PRESS_COUNT_LENGTH 1
00075 #define POLL_CONSTRAINT_RECORD_MAX_POLLING_KEY_PRESS_COUNT_OFFSET
00076     + POLL_CONSTRAINT_RECORD_OFFSET) (4
00076 #define POLL_CONSTRAINT_RECORD_MAX_POLLING_KEY_PRESS_COUNT_LENGTH 1
00077 #define POLL_CONSTRAINT_RECORD_MIN_POLLING_TIME_INTERVAL_OFFSET
00078     + POLL_CONSTRAINT_RECORD_OFFSET) (5
00078 #define POLL_CONSTRAINT_RECORD_MIN_POLLING_TIME_INTERVAL_LENGTH 4
00079 #define POLL_CONSTRAINT_RECORD_MAX_POLLING_TIME_INTERVAL_OFFSET
00080     + POLL_CONSTRAINT_RECORD_OFFSET) (9
00080 #define POLL_CONSTRAINT_RECORD_MAX_POLLING_TIME_INTERVAL_LENGTH 4
00081
00082 #define POLL_CONSTRAINT_RECORD_MIN_POLLING_TIME_INTERVAL_LOWER_BOUND
00083 #define POLL_CONSTRAINT_RECORD_MIN_POLLING_TIME_INTERVAL_UPPER_BOUND
00084     3600000
00084 #define POLL_CONSTRAINT_RECORD_MAX_POLLING_TIME_INTERVAL_LOWER_BOUND
00085     60000
00085 #define POLL_CONSTRAINT_RECORD_MAX_POLLING_TIME_INTERVAL_UPPER_BOUND
00086     8640000
00086
00087 // Poll configuration attribute related macros
00088 #define POLL_CONFIGURATION_POLLING_METHOD_ID_OFFSET 0
00089 #define POLL_CONFIGURATION_POLLING_METHOD_ID_LENGTH 1
00090 #define POLL_CONFIGURATION_POLLING_TRIGGER_CONFIG_OFFSET 1

```

```

00091 #define POLL_CONFIGURATION_POLLING_TRIGGER_CONFIG_LENGTH          2
00092 #define POLL_CONFIGURATION_POLLING_KEY_PRESS_COUNTER_OFFSET      3
00093 #define POLL_CONFIGURATION_POLLING_KEY_PRESS_COUNTER_LENGTH       1
00094 #define POLL_CONFIGURATION_POLLING_TIME_INTERVAL_OFFSET           4
00095 #define POLL_CONFIGURATION_POLLING_TIME_INTERVAL_LENGTH          4
00096 #define POLL_CONFIGURATION_POLLING_TIMEOUT_OFFSET                 8
00097 #define POLL_CONFIGURATION_POLLING_TIMEOUT_LENGTH                  1
00098
00099 // TODO: poll triggers and min/max values should be set in AppBuilder (only if
00100 // the node is a poll client, this values are meaningless for a poll server).
00101
00102 #define MIN_POLLING_KEY_PRESS_COUNTER           1
00103 #define MAX_POLLING_KEY_PRESS_COUNTER          255
00104 #define SUPPORTED_TRIGGERS
    (EMBER_AF_RF4CE_GDP_POLLING_TRIGGER_TIME_BASED_POLLING_ENABLED \
00105           |
    EMBER_AF_RF4CE_GDP_POLLING_TRIGGER_POLLING_ON_KEY_PRESS_ENABLED)
00106
00107 // By default we set the poll client to support the heartbeat method and to
00108 // support time-based and key press triggers.
00109 #define APL_POLL_CONSTRAINTS_DEFAULT          {0x01, /*number of methods supported
* */
00110
    EMBER_AF_RF4CE_GDP_POLLING_METHOD_HEARTBEAT, \
00111           LOW_BYTE(SUPPORTED_TRIGGERS),
00112           HIGH_BYTE(SUPPORTED_TRIGGERS),
00113           MIN_POLLING_KEY_PRESS_COUNTER,
00114           MAX_POLLING_KEY_PRESS_COUNTER,
00115
    BYTE_0(EMBER_AF_PLUGIN_RF4CE_GDP_MIN_POLLING_INTERVAL_MS), \
00116    BYTE_1(EMBER_AF_PLUGIN_RF4CE_GDP_MIN_POLLING_INTERVAL_MS), \
00117    BYTE_2(EMBER_AF_PLUGIN_RF4CE_GDP_MIN_POLLING_INTERVAL_MS), \
00118    BYTE_3(EMBER_AF_PLUGIN_RF4CE_GDP_MIN_POLLING_INTERVAL_MS), \
00119    BYTE_0(EMBER_AF_PLUGIN_RF4CE_GDP_MAX_POLLING_INTERVAL_MS), \
00120    BYTE_1(EMBER_AF_PLUGIN_RF4CE_GDP_MAX_POLLING_INTERVAL_MS), \
00121    BYTE_2(EMBER_AF_PLUGIN_RF4CE_GDP_MAX_POLLING_INTERVAL_MS), \
00122    BYTE_3(EMBER_AF_PLUGIN_RF4CE_GDP_MAX_POLLING_INTERVAL_MS) }
00123
00124 // Identification capabilities attribute related macros
00125 #define IDENTIFICATION_CAPABILITIES_RESERVED_MASK        0xF1
00126 #define IDENTIFICATION_CAPABILITIES_SUPPORT_FLASH_LIGHT_BIT 0x02
00127 #define IDENTIFICATION_CAPABILITIES_SUPPORT_MAKE_SOUND_BIT 0x04
00128 #define IDENTIFICATION_CAPABILITIES_SUPPORT_VIBRATE_BIT   0x08
00129
00130 // We maintain an info table for each GDP attribute defined in Table 9.
00131 typedef struct {
00132     uint8_t id;
00133     uint8_t size;
00134     uint8_t bitmask;
00135     uint16_t dimension; // only for arrayed attributes
00136 } EmAfRf4ceGdpAttributeDescriptor;
00137
00138 // Bitmask field definitions.
00139 #define ATTRIBUTE_HAS_REMOTE_GET_ACCESS_BIT                0x01
00140 #define ATTRIBUTE_HAS_REMOTE_SET_ACCESS_BIT                 0x02
00141 #define ATTRIBUTE_HAS_REMOTE_PUSH_ACCESS_BIT               0x04
00142 #define ATTRIBUTE_HAS_REMOTE_PULL_ACCESS_BIT              0x08
00143 #define ATTRIBUTE_IS_TWO_DIMENSIONAL_ARRAYED            0x10
00144
00145 // The following attributes are not currently included in the attribute struct,
00146 // they have no remote access and we treat them as constants, some of them
00147 // configurable from the plugin options.
00148 // - aplKeyExchangeTransferCount
00149 // - aplMaxPairingCandidates
00150 // - aplBindingRecipientValidationWaitTime

```

```

00151 // - aplBindingOriginatorValidationWaitTime
00152
00153 // Array attributes are in the range 0x90--0x9F for GDP and 0xC0--0xDF for the
00154 // other profiles. This means that all array attributes have bit 7 set; bit 5
00155 // clear; and bits 4 or 6 or both set. Note that this macro is just a tad not
00156 // safe because it uses the parameter attributeId twice.
00157 #define IS_ARRAY_ATTRIBUTE(attributeId) \
00158     (READBITS((attributeId), BIT(7) | BIT(5)) == BIT(7) \
00159     && READBITS((attributeId), BIT(4) | BIT(6)) != 0x00)
00160
00161 // Test related stuff
00162 #define GDP_ATTRIBUTE_ONE_DIMENSIONAL_ARRAY_TEST_SIZE           2
00163 #define GDP_ATTRIBUTE_ONE_DIMENSIONAL_ARRAY_TEST_DIMENSION        3
00164 #define GDP_ATTRIBUTE_TWO_DIMENSIONAL_ARRAY_TEST_SIZE             4
00165 #define GDP_ATTRIBUTE_TWO_DIMENSIONAL_ARRAY_TEST_FIRST_DIMENSION 4
00166 #define GDP_ATTRIBUTE_TWO_DIMENSIONAL_ARRAY_TEST_SECOND_DIMENSION 5
00167
00168 typedef struct {
00169     uint16_t gdpVersion;
00170     uint32_t gdpCapabilities;
00171     uint8_t powerStatus;
00172     uint8_t pollConstraints[API_GDP_POLL_CONSTRAINTS_SIZE];
00173     uint8_t pollConfiguration[API_GDP_POLL_CONFIGURATION_SIZE];
00174     uint16_t autoCheckValidationPeriod;
00175     uint16_t linkLostWaitTime;
00176     uint8_t identificationCapabilities;
00177
00178     // Test arrayed attributes
00179 #if defined(EMBER_SCRIPTED_TEST)
00180     uint16_t settableScalarTest1;
00181     uint16_t settableScalarTest2;
00182     uint16_t oneDimensionalTestAttribute[
00183         GDP_ATTRIBUTE_ONE_DIMENSIONAL_ARRAY_TEST_DIMENSION];
00184     uint32_t twoDimensionalTestAttribute[
00185         GDP_ATTRIBUTE_TWO_DIMENSIONAL_ARRAY_TEST_FIRST_DIMENSION][
00186         GDP_ATTRIBUTE_TWO_DIMENSIONAL_ARRAY_TEST_SECOND_DIMENSION];
00187 #endif
00188 } EmAfRf4ceGdpAttributes;
00189
00190 extern EmAfRf4ceGdpAttributes
00191     emAfRf4ceGdpLocalNodeAttributes;
00192 extern EmAfRf4ceGdpAttributes
00193     emAfRf4ceGdpRemoteNodeAttributes;
00194
00195 #define WRITE_ACCESS  true
00196 #define READ_ACCESS   false
00197
00198 void emAfRf4ceGdpClearRemoteAttributes(void);
00199
00200 void emAfRf4ceGdpGetOrSetAttribute(
00201     EmAfRf4ceGdpAttributes *attributes,
00202     uint8_t attrId,
00203     uint16_t entryId,
00204     bool isGet,
00205     uint8_t *val);
00206
00207 #define emAfRf4ceGdpSetLocalAttribute(attrId, entryId, val)
00208     emAfRf4ceGdpGetOrSetAttribute(&emAfRf4ceGdpLocalNodeAttributes,
00209                                     (attrId),
00210                                     (entryId),
00211                                     false,
00212                                     (uint8_t*) (val));
00213
00214 #define emAfRf4ceGdpGetLocalAttribute(attrId, entryId, val)
00215     emAfRf4ceGdpGetOrSetAttribute(&emAfRf4ceGdpLocalNodeAttributes,
00216                                     (attrId),
00217                                     (entryId),
00218                                     (uint8_t*) (val));

```

```

00212         true,
00213         (uint8_t*) (val))
00214 #define emAfRf4ceGdpSetRemoteAttribute(attrId, entryId, val)
00215         emAfRf4ceGdpGetOrSetAttribute(&emAfRf4ceGdpRemoteNodeAttributes,
00216                                         (attrId),
00217                                         (entryId),
00218                                         false,
00219                                         (uint8_t*) (val))
00220 #define emAfRf4ceGdpGetRemoteAttribute(attrId, entryId, val)
00221         emAfRf4ceGdpGetOrSetAttribute(&emAfRf4ceGdpRemoteNodeAttributes,
00222                                         (attrId),
00223                                         (entryId),
00224                                         true,
00225                                         (uint8_t*) (val))
00226 #endif //__RF4CE_GDP_ATTRIBUTES_H__
00227
00228
00229 #endif //__RF4CE_GDP_ATTRIBUTES_H__

```

8.8 rf4ce-gdp-identification-client.h File Reference

Functions

- void `emberAfRf4ceGdpIdentificationClientDetectedUserInteraction` (void)

8.9 rf4ce-gdp-identification-client.h

```

00001 // Copyright 2014 Silicon Laboratories, Inc.
00002
00003 #ifndef __RF4CE_GDP_IDENTIFICATION_CLIENT_H__
00004 #define __RF4CE_GDP_IDENTIFICATION_CLIENT_H__
00005
00043 void emberAfRf4ceGdpIdentificationClientDetectedUserInteraction
        (void);
00044
00045 #endif /* __RF4CE_GDP_IDENTIFICATION_CLIENT_H__ */
00046
00047 // END addtogroup

```

8.10 rf4ce-gdp-identification-server.h File Reference

Functions

- EmberStatus `emberAfRf4ceGdpIdentificationServerIdentify` (uint8_t pairingIndex, `EmberAfRf4ceGdpClientNotificationIdentifyFlags` flags, uint16_t timeS)

8.11 rf4ce-gdp-identification-server.h

```

00001 // Copyright 2014 Silicon Laboratories, Inc.
00002

```

```

00003 #ifndef __RF4CE_GDP_IDENTIFICATION_SERVER__
00004 #define __RF4CE_GDP_IDENTIFICATION_SERVER__
00005
00044 EmberStatus emberAfRf4ceGdpIdentificationServerIdentify
00045     (uint8_t pairingIndex,
00045      EmberAfRf4ceGdpClientNotificationIdentifyFlags
00046      flags,
00047      uint16_t timeS);
00048
00049 #endif /* __RF4CE_GDP_IDENTIFICATION_SERVER__ */
00050
00051 // END addtogroup

```

8.12 rf4ce-gdp-identification.h File Reference

Macros

- #define IDENTIFICATION PROCEDURE_DELAY_MSEC
- #define IDENTIFICATION PROCEDURE_AFTER_FAILURE_DELAY_SEC
- #define IDENTIFICATION PROCEDURE_CLIENT_MAX_RETIRES

Functions

- void emAfRf4ceGdpIdentificationNotifyBindingComplete (uint8_t pairingIndex)
- void emAfRf4ceGdpIncomingIdentifyCallback (EmberAfRf4ceGdpClientNotificationIdentifyFlags
flags, uint16_t timeS)

8.12.1 Macro Definition Documentation

8.12.1.1 #define IDENTIFICATION PROCEDURE_DELAY_MSEC

Definition at line 6 of file rf4ce-gdp-identification.h.

8.12.1.2 #define IDENTIFICATION PROCEDURE_AFTER_FAILURE_DELAY_SEC

Definition at line 10 of file rf4ce-gdp-identification.h.

8.12.1.3 #define IDENTIFICATION PROCEDURE_CLIENT_MAX_RETIRES

Definition at line 14 of file rf4ce-gdp-identification.h.

8.12.2 Function Documentation

8.12.2.1 void emAfRf4ceGdpIdentificationNotifyBindingComplete (uint8_t pairingIndex)

8.12.2.2 void emAfRf4ceGdpIncomingIdentifyCallback (EmberAfRf4ceGdpClientNotificationIdentify- Flags flags, uint16_t timeS)

8.13 rf4ce-gdp-identification.h

```

00001 // Copyright 2014 Silicon Laboratories, Inc.
00002
00003 // We delay the procedure in case the enhanced security procedure or the
00004 // polling
00005 // negotiation procedure take place, which should happen after binding has
00006 // completed.
00006 #define IDENTIFICATION_PROCEDURE_DELAY_MSEC      500
00007
00008 // If the identification procedure fails, we wait a longer delay before trying
00009 // again.
00010 #define IDENTIFICATION_PROCEDURE_AFTER_FAILURE_DELAY_SEC    60
00011
00012 // This defines how many times the identification procedure is started again
00013 // after it failed.
00014 #define IDENTIFICATION_PROCEDURE_CLIENT_MAX_RETIRES      3
00015
00016
00017 extern void emAfRf4ceGdpIdentificationNotifyBindingComplete
00018     (uint8_t pairingIndex);
00019 extern void emAfRf4ceGdpIncomingIdentifyCallback
00020     (EmberAfRf4ceGdpClientNotificationIdentifyFlags
00021         flags,
00020                     uint16_t timeS);
00021

```

8.14 rf4ce-gdp-internal.h File Reference

Data Structures

- struct [EmAfGdpPairingCandidate](#)
- struct [EmAfDiscoveryOrPairrequestData](#)
- struct [EmAfBindingInfo](#)

Macros

- #define [NWK_DISCOVERY_REPEATITION_INTERVAL_MS](#)
- #define [NWK_MAX_DISCOVERY_REPEATITIONS](#)
- #define [NWK_MAX_REPORTED_NODE_DESCRIPTOROS](#)
- #define [NWK_DISCOVERY_LQI_THRESHOLD](#)
- #define [APL_MAX_PAIRING_CANDIDATES](#)
- #define [APLC_CONFIG_BLACKOUT_TIME_MS](#)
- #define [BLACKOUT_TIME_DELTA_MS](#)
- #define [BLACKOUT_TIME_ORIGINATOR_MS](#)
- #define [BLACKOUT_TIME_RECIPIENT_MS](#)
- #define [APLC_BIND_WINDOW_DURATION_MS](#)
- #define [APLC_MAX_AUTO_CHECK_VALIDATION_PERIOD_MS](#)
- #define [APLC_MAX_CONFIG_WAIT_TIME_MS](#)
- #define [APLC_MAX_NORMAL_VALIDATION_DURATION_MS](#)
- #define [APLC_MAX_EXTENDED_VALIDATION_DURATION_MS](#)
- #define [APLC_MAX_POLLING_TIMEOUT_MS](#)
- #define [APLC_MAX_RX_ON_WAIT_TIME_MS](#)
- #define [APLC_MAX_RESPONSE_WAIT_TIME_MS](#)
- #define [APLC_MIN_KEY_EXCHANGE_TRANSFER_COUNT](#)
- #define [GDP_HEADER_LENGTH](#)
- #define [GDP_HEADER_FRAME_CONTROL_OFFSET](#)

- #define GDP_HEADER_FRAME_CONTROL_COMMAND_CODE_MASK
- #define GDP_HEADER_FRAME_CONTROL_GDP_COMMAND_FRAME_MASK
- #define GDP_HEADER_FRAME_CONTROL_DATA_PENDING_MASK
- #define GDP_PAYLOAD_OFFSET
- #define GENERIC_RESPONSE_LENGTH
- #define GENERIC_RESPONSE_RESPONSE_CODE_OFFSET
- #define CONFIGURATION_COMPLETE_LENGTH
- #define CONFIGURATION_COMPLETE_STATUS_OFFSET
- #define HEARTBEAT_LENGTH
- #define HEARTBEAT_TRIGGER_OFFSET
- #define GET_ATTRIBUTES_LENGTH
- #define GET_ATTRIBUTES_RESPONSE_LENGTH
- #define PUSH_ATTRIBUTES_LENGTH
- #define SET_ATTRIBUTES_LENGTH
- #define PULL_ATTRIBUTES_LENGTH
- #define PULL_ATTRIBUTES_RESPONSE_LENGTH
- #define CHECK_VALIDATION_LENGTH
- #define CHECK_VALIDATION_SUBTYPE_OFFSET
- #define CHECK_VALIDATION_PAYLOAD_OFFSET
- #define CHECK_VALIDATION_SUBTYPE_REQUEST_LENGTH
- #define CHECK_VALIDATION_SUBTYPE_REQUEST_CONTROL_OFFSET
- #define CHECK_VALIDATION_SUBTYPE_RESPONSE_LENGTH
- #define CHECK_VALIDATION_SUBTYPE_RESPONSE_STATUS_OFFSET
- #define CLIENT_NOTIFICATION_LENGTH
- #define CLIENT_NOTIFICATION_SUBTYPE_OFFSET
- #define CLIENT_NOTIFICATION_PAYLOAD_OFFSET
- #define CLIENT_NOTIFICATION_SUBTYPE_IDENTIFY_LENGTH
- #define CLIENT_NOTIFICATION_SUBTYPE_IDENTIFY_PAYLOAD_LENGTH
- #define CLIENT_NOTIFICATION_SUBTYPE_IDENTIFY_PAYLOAD_FLAGS_OFFSET
- #define CLIENT_NOTIFICATION_SUBTYPE_IDENTIFY_PAYLOAD_TIME_OFFSET
- #define CLIENT_NOTIFICATION_SUBTYPE_REQUEST_POLL_NEGOTIATION_LENGTH
- #define CLIENT_NOTIFICATION_SUBTYPE_REQUEST_POLL_NEGOTIATION_PAYLOAD_LENGTH
- #define KEY_EXCHANGE_LENGTH
- #define KEY_EXCHANGE_SUBTYPE_OFFSET
- #define KEY_EXCHANGE_PAYLOAD_OFFSET
- #define KEY_EXCHANGE_SUBTYPE_CHALLENGE_LENGTH
- #define KEY_EXCHANGE_SUBTYPE_CHALLENGE_FLAGS_OFFSET
- #define KEY_EXCHANGE_SUBTYPE_CHALLENGE RAND_A_OFFSET
- #define KEY_EXCHANGE_SUBTYPE_CHALLENGE_RESPONSE_LENGTH
- #define KEY_EXCHANGE_SUBTYPE_CHALLENGE_RESPONSE_FLAGS_OFFSET
- #define KEY_EXCHANGE_SUBTYPE_CHALLENGE_RESPONSE RAND_B_OFFSET
- #define KEY_EXCHANGE_SUBTYPE_CHALLENGE_RESPONSE_TAG_B_OFFSET
- #define KEY_EXCHANGE_SUBTYPE_RESPONSE_LENGTH
- #define KEY_EXCHANGE_SUBTYPE_RESPONSE_TAG_A_OFFSET
- #define KEY_EXCHANGE_SUBTYPE_CONFIRM_LENGTH
- #define COMMAND_CODE_MAXIMUM
- #define GDP_CAPABILITIES_SUPPORT_EXTENDED_VALIDATION_BIT
- #define GDP_CAPABILITIES_SUPPORT_EXTENDED_VALIDATION_OFFSET
- #define GDP_CAPABILITIES_SUPPORT_POLL_SERVER_BIT

- #define GDP_CAPABILITIES_SUPPORT_POLL_SERVER_OFFSET
- #define GDP_CAPABILITIES_SUPPORT_POLL_CLIENT_BIT
- #define GDP_CAPABILITIES_SUPPORT_POLL_CLIENT_OFFSET
- #define GDP_CAPABILITIES_SUPPORT_IDENTIFICATION_SERVER_BIT
- #define GDP_CAPABILITIES_SUPPORT_IDENTIFICATION_SERVER_OFFSET
- #define GDP_CAPABILITIES_SUPPORT_IDENTIFICATION_CLIENT_BIT
- #define GDP_CAPABILITIES_SUPPORT_IDENTIFICATION_CLIENT_OFFSET
- #define GDP_CAPABILITIES_SUPPORT_ENHANCED_SECURITY_BIT
- #define GDP_CAPABILITIES_SUPPORT_ENHANCED_SECURITY_OFFSET
- #define GDP_CAPABILITIES_SUPPORT_SHARED_SECRET_OF_LOCAL_VENDOR_BIT
- #define GDP_CAPABILITIES_SUPPORT_SHARED_SECRET_OF_LOCAL_VENDOR_OFFSET
- #define GDP_CAPABILITIES_SUPPORT_SHARED_SECRET_OF_REMOTE_VENDOR_BIT
- #define GDP_CAPABILITIES_SUPPORT_SHARED_SECRET_OF_REMOTE_VENDOR_OFFSET
- #define GDP_STANDARD_SHARED_SECRET
- #define GDP_SHARED_SECRET_SIZE
- #define USER_STRING_APPLICATION_SPECIFIC_USER_STRING_OFFSET
- #define USER_STRING_APPLICATION_SPECIFIC_USER_STRING_LENGTH
- #define USER_STRING_NULL_BYTE_OFFSET
- #define USER_STRING_DISC_REQUEST_VENDOR_ID_FILTER_OFFSET
- #define USER_STRING_DISC_REQUEST_VENDOR_ID_FILTER_LENGTH
- #define USER_STRING_DISC_REQUEST_MIN_MAX_CLASS_FILTER_OFFSET
- #define USER_STRING_DISC_REQUEST_MIN_MAX_CLASS_FILTER_MIN_CLASS_NUM_MASK
- #define USER_STRING_DISC_REQUEST_MIN_MAX_CLASS_FILTER_MIN_CLASS_NUM_OFFSET
- #define USER_STRING_DISC_REQUEST_MIN_MAX_CLASS_FILTER_MAX_CLASS_NUM_MASK
- #define USER_STRING_DISC_REQUEST_MIN_MAX_CLASS_FILTER_MAX_CLASS_NUM_OFFSET
- #define USER_STRING_DISC_REQUEST_MIN_LQI_FILTER_OFFSET
- #define USER_STRING_DISC_REQUEST_RESERVED_BYTES_OFFSET
- #define USER_STRING_DISC_REQUEST_RESERVED_BYTES_LENGTH
- #define USER_STRING_DISC_RESPONSE_RESERVED_BYTES_OFFSET
- #define USER_STRING_DISC_RESPONSE_RESERVED_BYTES_LENGTH
- #define USER_STRING_DISC_RESPONSE_TERTIARY_CLASS_DESCRIPTOR_OFFSET
- #define USER_STRING_DISC_RESPONSE_SECONDARY_CLASS_DESCRIPTOR_OFFSET
- #define USER_STRING_DISC_RESPONSE_PRIMARY_CLASS_DESCRIPTOR_OFFSET
- #define USER_STRING_DISC_RESPONSE_DISCOVERY_LQI_THRESHOLD_OFFSET
- #define USER_STRING_PAIR_REQUEST_ADVANCED_BINDING_SUPPORT_OFFSET
- #define USER_STRING_PAIR_REQUEST_RESERVED_BYTES_OFFSET
- #define USER_STRING_PAIR_REQUEST_RESERVED_BYTES_LENGTH
- #define ADVANCED_BINDING_SUPPORT_FIELD_BINDING_PROXY_SUPPORTED_BIT
- #define CLASS_DESCRIPTOR_NUMBER_MASK
- #define CLASS_DESCRIPTOR_NUMBER_OFFSET
- #define CLASS_DESCRIPTOR_DUPLICATE_HANDLING_MASK
- #define CLASS_DESCRIPTOR_DUPLICATE_HANDLING_OFFSET
- #define CLASS_DESCRIPTOR_RESERVED_MASK
- #define CLASS_NUMBER_PRE_COMMISSIONED

- #define CLASS_NUMBER_BUTTON_PRESS_INDICATION
- #define CLASS_NUMBER_DISCOVERABLE_ONLY
- #define CLASS_DESCRIPTOR_DUPLICATE_HANDLING_AS_IS
- #define CLASS_DESCRIPTOR_DUPLICATE_HANDLING_RECLASSIFY
- #define CLASS_DESCRIPTOR_DUPLICATE_HANDLING_ABORT
- #define CLASS_DESCRIPTOR_DUPLICATE_HANDLING_RESERVED
- #define AS_IS
- #define RECLASSIFY
- #define ABORT
- #define GDP_VERSION_NONE
- #define GDP_VERSION_1_X
- #define GDP_VERSION_2_0
- #define GDP_1_X_BASED_PROFILE_ID_LIST
- #define GDP_1_X_BASED_PROFILE_ID_LIST_LENGTH
- #define GDP_2_0_BASED_PROFILE_ID_LIST
- #define GDP_2_0_BASED_PROFILE_ID_LIST_LENGTH
- #define PUBLIC_STATE_MASK
- #define INTERNAL_STATE_MASK
- #define INTERNAL_STATE_OFFSET
- #define INTERNAL_STATE_NONE
- #define INTERNAL_STATE_ORIGINATOR_GDP_CONFIG_PUSH_PENDING
- #define INTERNAL_STATE_ORIGINATOR_GDP_CONFIG_GET_PENDING
- #define INTERNAL_STATE_ORIGINATOR_GDP_CONFIG_PULL_PENDING
- #define INTERNAL_STATE_ORIGINATOR_GDP_CONFIG_COMPLETE_PENDING
- #define INTERNAL_STATE_ORIGINATOR_GDP_PROFILES_CONFIG
- #define INTERNAL_STATE_ORIGINATOR_GDP_VALIDATION
- #define INTERNAL_STATE_ORIGINATOR_GDP_KEY_EXCHANGE_BLACKOUT_PENDING
- #define INTERNAL_STATE_RECIPIENT_GDP_STACK_STATUS_NETWORK_UP_PENDING
- #define INTERNAL_STATE_RECIPIENT_GDP_RESTORE_PAIRING_ENTRY_PENDING
- #define INTERNAL_STATE_RECIPIENT_GDP_CONFIG_WAITING_FOR_PUSH
- #define INTERNAL_STATE_RECIPIENT_GDP_CONFIG_WAITING_FOR_GET
- #define INTERNAL_STATE_RECIPIENT_GDP_CONFIG_WAITING_FOR_PULL
- #define INTERNAL_STATE_RECIPIENT_GDP_CONFIG_WAITING_FOR_COMPLETE
- #define INTERNAL_STATE_RECIPIENT_GDP_PROFILES_CONFIG
- #define INTERNAL_STATE_RECIPIENT_GDP_VALIDATION
- #define INTERNAL_STATE_GDP_SECURITY_KEY_CHALLENGE_PENDING
- #define INTERNAL_STATE_GDP_SECURITY_KEY_CHALLENGE_RESPONSE_PENDING
- #define INTERNAL_STATE_GDP_SECURITY_KEY_RESPONSE_PENDING
- #define INTERNAL_STATE_GDP_POLL_CONFIG_CLIENT_PUSH_PENDING
- #define INTERNAL_STATE_GDP_POLL_CONFIG_CLIENT_PULL_PENDING
- #define INTERNAL_STATE_GDP_IDENTIFICATION_CLIENT_PUSH_PENDING
- #define INTERNAL_STATE_GDP_POLL_CLIENT_HEARTBEAT_PENDING
- #define publicBindState()
- #define internalGdpState()
- #define setPublicState(state, init)
- #define setInternalState(state)
- #define isInternalStateBindingOriginator()
- #define isInternalStateBindingRecipient()
- #define isInternalStateSecurity()

- #define `isInternalStatePollNegotiation()`
- #define `isInternalStateIdentification()`
- #define `CANDIDATE_INFO_ENTRY_IN_USE_BIT`
- #define `CANDIDATE_INFO_ENTRY_IN_USE_OFFSET`
- #define `CANDIDATE_INFO_PAIRING_ATTEMPTED_BIT`
- #define `CANDIDATE_INFO_PAIRING_ATTEMPTED_OFFSET`
- #define `CANDIDATE_INFO_PROXY_CANDIDATE_BIT`
- #define `CANDIDATE_INFO_PROXY_CANDIDATE_OFFSET`
- #define `PAIRING_ENTRY_BINDING_STATUS_MASK`
- #define `PAIRING_ENTRY_BINDING_STATUS_OFFSET`
- #define `PAIRING_ENTRY_BINDING_STATUS_NOT_BOUND`
- #define `PAIRING_ENTRY_BINDING_STATUS_BOUND_ORIGINATOR`
- #define `PAIRING_ENTRY_BINDING_STATUS_BOUND_RECIPIENT`
- #define `PAIRING_ENTRY_BINDING_COMPLETE_BIT`
- #define `PAIRING_ENTRY_BINDING_COMPLETE_OFFSET`
- #define `PAIRING_ENTRY_POLLING_ACTIVE_BIT`
- #define `PAIRING_ENTRY_POLLING_ACTIVE_OFFSET`
- #define `PAIRING_ENTRY_IDENTIFICATION_ACTIVE_BIT`
- #define `PAIRING_ENTRY_IDENTIFICATION_ACTIVE_OFFSET`
- #define `PAIRING_ENTRY_REMOTE_NODE_SUPPORTS_ENHANCED_SECURITY_BIT`
- #define `PAIRING_ENTRY_REMOTE_NODE_SUPPORTS_ENHANCED_SECURITY_OFFSET`
- #define `PAIRING_ENTRY_REMOTE_NODE_SUPPORTS_IDENTIFICATION_BIT`
- #define `PAIRING_ENTRY_REMOTE_NODE_SUPPORTS_IDENTIFICATION_OFFSET`
- #define `PAIRING_ENTRY_REMOTE_NODE_SUPPORTS_POLLING_BIT`
- #define `PAIRING_ENTRY_REMOTE_NODE_SUPPORTS_POLLING_OFFSET`
- #define `emAfRf4ceGdpIsBusy()`
- #define `GDP_SET_VALUE_BINDING_ORIGINATOR_PARAMETERS_BYTES_LENGTH`
- #define `GDP_SET_VALUE_BINDING_RECIPIENT_PARAMETERS_BYTES_LENGTH`
- #define `GDP_SET_VALUE_FLAG_LENGTH`
- #define `emAfRf4ceGdpOutgoingCommandsSetPendingFlag()`
- #define `emAfRf4ceGdpOutgoingCommandsClearPendingFlag()`
- #define `debugDiscoveryResponseDrop(reason)`
- #define `debugCandidateAdded(reason)`
- #define `debugScriptCheck(reason)`

Functions

- bool `emAfIsProfileGdpBased` (uint8_t profileId, uint8_t gdpCheckVersion)
- bool `emAfRf4ceIsProfileSupported` (uint8_t profileId, const uint8_t *profileIdList, uint8_t profileIdListLength)
- uint8_t `emAfCheckDeviceTypeAndProfileIdMatch` (uint8_t checkDeviceType, uint8_t *compareDevTypeList, uint8_t compareDevTypeListLength, uint8_t *checkProfileIdList, uint8_t checkProfileIdListLength, uint8_t *compareProfileIdList, uint8_t compareProfileIdListLength, uint8_t *matchingProfileIdList)
- bool `emAfRf4ceGdpMaybeStartNextProfileSpecificConfigurationProcedure` (bool isOriginator, const uint8_t *remoteProfileIdList, uint8_t remoteProfileIdListLength)
- void `emAfRf4ceGdpNotifyBindingCompleteToProfiles` (`EmberAfRf4ceGdpBindingStatus` status, uint8_t pairingIndex, const uint8_t *remoteProfileIdList, uint8_t remoteProfileIdListLength)
- EmberStatus `emAfRf4ceGdpInitiateKeyExchangeInternal` (uint8_t pairingIndex, bool intCall)
- void `emAfRf4ceGdpNoteProfileSpecificConfigurationStart` (void)

- void `emAfGdpAddToProfileIdList` (uint8_t *srcProfileIdList, uint8_t srcProfileIdListLength, EmberRf4ceApplicationInfo *destAppInfo, uint8_t gdpVersion)
- uint8_t `emAfRf4ceGdpGetGdpVersion` (const uint8_t *profileIdList, uint8_t profileIdListLength)
- void `emAfRf4ceGdpRecipientInitCallback` (void)
- void `emAfRf4ceGdpOriginatorStackStatusCallback` (EmberStatus status)
- void `emAfRf4ceGdpRecipientStackStatusCallback` (EmberStatus status)
- void `emAfRf4ceGdpUpdatePublicStatus` (bool init)
- uint8_t `emAfRf4ceGdpGetPairingBindStatus` (uint8_t pairingIndex)
- void `emAfRf4ceGdpSetPairingBindStatus` (uint8_t pairingIndex, uint8_t status)
- void `emAfRf4ceGdpGetPairingKey` (uint8_t pairingIndex, EmberKeyData *key)
- void `emAfRf4ceGdpSetPairingKey` (uint8_t pairingIndex, EmberKeyData *key)
- bool `emAfRf4ceGdpSecurityGetRandomString` (EmberAfRf4ceGdpRand *rn)
- void `emAfRf4ceGdpSecurityValidationCompleteCallback` (uint8_t pairingIndex)
- void `emAfRf4ceGdpAttributesInitCallback` (void)
- void `emAfRf4ceGdpIncomingGenericResponse` (EmberAfRf4ceGdpResponseCode responseCode)
- void `emAfRf4ceGdpIncomingConfigurationComplete` (EmberAfRf4ceGdpStatus status)
- void `emAfRf4ceGdpIncomingHeartbeat` (EmberAfRf4ceGdpHeartbeatTrigger trigger)
- void `emAfRf4ceGdpIncomingGetAttributes` (void)
- void `emAfRf4ceGdpIncomingGetAttributesResponse` (void)
- void `emAfRf4ceGdpIncomingPushAttributes` (void)
- void `emAfRf4ceGdpIncomingSetAttributes` (void)
- void `emAfRf4ceGdpIncomingPullAttributes` (void)
- void `emAfRf4ceGdpIncomingPullAttributesResponse` (void)
- void `emAfRf4ceGdpIncomingCheckValidationRequest` (uint8_t control)
- void `emAfRf4ceGdpIncomingCheckValidationResponse` (EmberAfRf4ceGdpCheckValidationStatus status)
- void `emAfRf4ceGdpCheckValidationResponseSent` (EmberStatus status)
- void `emAfRf4ceGdpHeartbeatSent` (EmberStatus status)
- void `emAfRf4ceGdpIncomingKeyExchangeChallenge` (EmberAfRf4ceGdpKeyExchangeFlags flags, const EmberAfRf4ceGdpRand *randA)
- void `emAfRf4ceGdpIncomingKeyExchangeChallengeResponse` (EmberAfRf4ceGdpKeyExchangeFlags flags, const EmberAfRf4ceGdpRand *randB, const EmberAfRf4ceGdpTag *tagB)
- void `emAfRf4ceGdpIncomingKeyExchangeResponse` (const EmberAfRf4ceGdpTag *tagA)
- void `emAfRf4ceGdpKeyExchangeResponseSent` (EmberStatus status)
- void `emAfRf4ceGdpIncomingKeyExchangeConfirm` (bool secured)
- void `emAfRf4ceGdpIncomingClientNotification` (EmberAfRf4ceGdpClientNotificationSubtype subType, const uint8_t *clientNotificationPayload, uint8_t clientNotificationPayloadLength)
- void `emAfRf4ceZrcIncomingGenericResponse` (EmberAfRf4ceGdpResponseCode responseCode)
- void `emAfRf4ceZrcIncomingConfigurationComplete` (EmberAfRf4ceGdpStatus status)
- void `emAfRf4ceZrcIncomingGetAttributes` (void)
- void `emAfRf4ceZrcIncomingGetAttributesResponse` (void)
- void `emAfRf4ceZrcIncomingPushAttributes` (void)
- void `emAfRf4ceZrcIncomingSetAttributes` (void)
- void `emAfRf4ceZrcIncomingPullAttributes` (void)
- void `emAfRf4ceZrcIncomingPullAttributesResponse` (void)
- void `emAfRf4ceZrcIncomingClientNotification` (EmberAfRf4ceGdpClientNotificationSubtype subType, const uint8_t *clientNotificationPayload, uint8_t clientNotificationPayloadLength)
- bool `emAfRf4ceGdpHasAttributeRecord` (void)
- bool `emAfRf4ceGdpAppendAttributeIdentificationRecord` (const EmberAfRf4ceGdpAttributeIdentificationRecord *record)

- bool `emAfRf4ceGdpFetchAttributeIdentificationRecord` (`EmberAfRf4ceGdpAttributeIdentificationRecord` *record)
- bool `emAfRf4ceGdpAppendAttributeStatusRecord` (const `EmberAfRf4ceGdpAttributeStatusRecord` *record)
- bool `emAfRf4ceGdpFetchAttributeStatusRecord` (`EmberAfRf4ceGdpAttributeStatusRecord` *record)
- bool `emAfRf4ceGdpAppendAttributeRecord` (const `EmberAfRf4ceGdpAttributeRecord` *record)
- bool `emAfRf4ceGdpFetchAttributeRecord` (`EmberAfRf4ceGdpAttributeRecord` *record)
- void `emAfRf4ceGdpResetFetchAttributeFinger` (void)
- void `emAfRf4ceGdpStartAttributesCommand` (`EmberAfRf4ceGdpCommandCode` commandCode)
- EmberStatus `emAfRf4ceGdpSendAttributesCommand` (uint8_t pairingIndex, uint8_t profileId, uint16_t vendorId)
- EmberStatus `emAfRf4ceGdpSetDiscoveryResponseAppInfo` (bool pushButton, uint8_t gdpVersion)
- EmberStatus `emAfRf4ceGdpSetPairResponseAppInfo` (const `EmberRf4ceApplicationInfo` *pairRequestAppInfo)
- EmberStatus `emAfRf4ceGdpSendProfileSpecificCommand` (uint8_t pairingIndex, uint8_t profileId, uint16_t vendorId, EmberRf4ceTxOption txOptions, uint8_t commandId, uint8_t *commandPayload, uint8_t commandPayloadLength, uint8_t *messageTag)
- EmberStatus `emAfRf4ceGdpKeyExchangeChallenge` (uint8_t pairingIndex, uint16_t vendorId, `EmberAfRf4ceGdpKeyExchangeFlags` flags, const `EmberAfRf4ceGdpRand` *randA)
- EmberStatus `emAfRf4ceGdpKeyExchangeChallengeResponse` (uint8_t pairingIndex, uint16_t vendorId, `EmberAfRf4ceGdpKeyExchangeFlags` flags, const `EmberAfRf4ceGdpRand` *randB, const `EmberAfRf4ceGdpTag` *tagB)
- EmberStatus `emAfRf4ceGdpKeyExchangeResponse` (uint8_t pairingIndex, uint16_t vendorId, const `EmberAfRf4ceGdpTag` *tagA)
- EmberStatus `emAfRf4ceGdpKeyExchangeConfirm` (uint8_t pairingIndex, uint16_t vendorId)
- void `emAfRf4ceGdpSetPushButtonPendingReceivedFlag` (bool set)
- void `emAfRf4ceGdpSetProxyBindingFlag` (bool set)
- void `emAfGdpStartBlackoutTimer` (uint8_t state)
- void `emAfGdpStartCommandPendingTimer` (uint8_t state, uint16_t timeMs)

Variables

- const uint8_t `emAfRf4ceGdpApplicationSpecificUserString` [`USER_STRING_APPLICATION_SPECIFIC_USER_STRING_LENGTH`]
- const uint8_t `emAfGdp1xProfiles` []
- const uint8_t `emAfGdp20Profiles` []
- uint8_t `emAfTemporaryPairingIndex`
- uint8_t `emAfCurrentProfileSpecificIndex`
- uint16_t `emAfGdpState`
- uint32_t `emberAfPluginRf4ceGdpCapabilities`
- EmberEventControl `emberAfPluginRf4ceGdpPendingCommandEventControl`
- EmberEventControl `emberAfPluginRf4ceGdpBlackoutTimeEventControl`
- EmberEventControl `emberAfPluginRf4ceGdpValidationEventControl`
- `EmAfBindingInfo` `emAfGdpPeerInfo`
- uint8_t `emAfRf4ceGdpOutgoingCommandFrameControl`

8.14.1 Macro Definition Documentation

8.14.1.1 #define NWK_DISCOVERY_REPEATITION_INTERVAL_MS

Definition at line 3 of file `rf4ce-gdp-internal.h`.

8.14.1.2 #define NWK_MAX_DISCOVERY_REPEATITIONS

Definition at line 4 of file [rf4ce-gdp-internal.h](#).

8.14.1.3 #define NWK_MAX_REPORTED_NODE_DESCRIPTOROS

Definition at line 5 of file [rf4ce-gdp-internal.h](#).

8.14.1.4 #define NWK_DISCOVERY_LQI_THRESHOLD

Definition at line 9 of file [rf4ce-gdp-internal.h](#).

8.14.1.5 #define APL_MAX_PAIRING_CANDIDATES

Definition at line 14 of file [rf4ce-gdp-internal.h](#).

8.14.1.6 #define APLC_CONFIG_BLACKOUT_TIME_MS

Definition at line 19 of file [rf4ce-gdp-internal.h](#).

8.14.1.7 #define BLACKOUT_TIME_DELTA_MS

Definition at line 21 of file [rf4ce-gdp-internal.h](#).

8.14.1.8 #define BLACKOUT_TIME_ORIGINATOR_MS

Definition at line 25 of file [rf4ce-gdp-internal.h](#).

8.14.1.9 #define BLACKOUT_TIME_RECIPIENT_MS

Definition at line 28 of file [rf4ce-gdp-internal.h](#).

8.14.1.10 #define APLC_BIND_WINDOW_DURATION_MS

Definition at line 33 of file [rf4ce-gdp-internal.h](#).

8.14.1.11 #define APLC_MAX_AUTO_CHECK_VALIDATION_PERIOD_MS

Definition at line 36 of file [rf4ce-gdp-internal.h](#).

8.14.1.12 #define APLC_MAX_CONFIG_WAIT_TIME_MS

Definition at line 40 of file [rf4ce-gdp-internal.h](#).

8.14.1.13 #define APLC_MAX_NORMAL_VALIDATION_DURATION_MS

Definition at line 44 of file [rf4ce-gdp-internal.h](#).

8.14.1.14 #define APLC_MAX_EXTENDED_VALIDATION_DURATION_MS

Definition at line 48 of file [rf4ce-gdp-internal.h](#).

8.14.1.15 #define APLC_MAX_POLLING_TIMEOUT_MS

Definition at line 52 of file [rf4ce-gdp-internal.h](#).

8.14.1.16 #define APLC_MAX_RX_ON_WAIT_TIME_MS

Definition at line 57 of file [rf4ce-gdp-internal.h](#).

8.14.1.17 #define APLC_MAX_RESPONSE_WAIT_TIME_MS

Definition at line 61 of file [rf4ce-gdp-internal.h](#).

8.14.1.18 #define APLC_MIN_KEY_EXCHANGE_TRANSFER_COUNT

Definition at line 65 of file [rf4ce-gdp-internal.h](#).

8.14.1.19 #define GDP_HEADER_LENGTH

Definition at line 69 of file [rf4ce-gdp-internal.h](#).

8.14.1.20 #define GDP_HEADER_FRAME_CONTROL_OFFSET

Definition at line 70 of file [rf4ce-gdp-internal.h](#).

8.14.1.21 #define GDP_HEADER_FRAME_CONTROL_COMMAND_CODE_MASK

Definition at line 71 of file [rf4ce-gdp-internal.h](#).

8.14.1.22 #define GDP_HEADER_FRAME_CONTROL_GDP_COMMAND_FRAME_MASK

Definition at line 72 of file [rf4ce-gdp-internal.h](#).

8.14.1.23 #define GDP_HEADER_FRAME_CONTROL_DATA_PENDING_MASK

Definition at line 73 of file [rf4ce-gdp-internal.h](#).

8.14.1.24 #define GDP_PAYLOAD_OFFSET

Definition at line 74 of file [rf4ce-gdp-internal.h](#).

8.14.1.25 #define GENERIC_RESPONSE_LENGTH

Definition at line 78 of file [rf4ce-gdp-internal.h](#).

8.14.1.26 #define GENERIC_RESPONSE_RESPONSE_CODE_OFFSET

Definition at line 79 of file [rf4ce-gdp-internal.h](#).

8.14.1.27 #define CONFIGURATION_COMPLETE_LENGTH

Definition at line 83 of file [rf4ce-gdp-internal.h](#).

8.14.1.28 #define CONFIGURATION_COMPLETE_STATUS_OFFSET

Definition at line 84 of file [rf4ce-gdp-internal.h](#).

8.14.1.29 #define HEARTBEAT_LENGTH

Definition at line 88 of file [rf4ce-gdp-internal.h](#).

8.14.1.30 #define HEARTBEAT_TRIGGER_OFFSET

Definition at line 89 of file [rf4ce-gdp-internal.h](#).

8.14.1.31 #define GET_ATTRIBUTES_LENGTH

Definition at line 96 of file [rf4ce-gdp-internal.h](#).

8.14.1.32 #define GET_ATTRIBUTES_RESPONSE_LENGTH

Definition at line 106 of file [rf4ce-gdp-internal.h](#).

8.14.1.33 #define PUSH_ATTRIBUTES_LENGTH

Definition at line 115 of file [rf4ce-gdp-internal.h](#).

8.14.1.34 #define SET_ATTRIBUTES_LENGTH

Definition at line 124 of file [rf4ce-gdp-internal.h](#).

8.14.1.35 #define PULL_ATTRIBUTES_LENGTH

Definition at line 131 of file [rf4ce-gdp-internal.h](#).

8.14.1.36 #define PULL_ATTRIBUTES_RESPONSE_LENGTH

Definition at line 141 of file [rf4ce-gdp-internal.h](#).

8.14.1.37 #define CHECK_VALIDATION_LENGTH

Definition at line 146 of file [rf4ce-gdp-internal.h](#).

8.14.1.38 #define CHECK_VALIDATION_SUBTYPE_OFFSET

Definition at line 147 of file [rf4ce-gdp-internal.h](#).

8.14.1.39 #define CHECK_VALIDATION_PAYLOAD_OFFSET

Definition at line 148 of file [rf4ce-gdp-internal.h](#).

8.14.1.40 #define CHECK_VALIDATION_SUBTYPE_REQUEST_LENGTH

Definition at line 149 of file [rf4ce-gdp-internal.h](#).

8.14.1.41 #define CHECK_VALIDATION_SUBTYPE_REQUEST_CONTROL_OFFSET

Definition at line 150 of file [rf4ce-gdp-internal.h](#).

8.14.1.42 #define CHECK_VALIDATION_SUBTYPE_RESPONSE_LENGTH

Definition at line 151 of file [rf4ce-gdp-internal.h](#).

8.14.1.43 #define CHECK_VALIDATION_SUBTYPE_RESPONSE_STATUS_OFFSET

Definition at line 152 of file [rf4ce-gdp-internal.h](#).

8.14.1.44 #define CLIENT_NOTIFICATION_LENGTH

Definition at line 157 of file [rf4ce-gdp-internal.h](#).

8.14.1.45 #define CLIENT_NOTIFICATION_SUBTYPE_OFFSET

Definition at line 158 of file [rf4ce-gdp-internal.h](#).

8.14.1.46 #define CLIENT_NOTIFICATION_PAYLOAD_OFFSET

Definition at line 159 of file [rf4ce-gdp-internal.h](#).

8.14.1.47 #define CLIENT_NOTIFICATION_SUBTYPE_IDENTITY_LENGTH

Definition at line 160 of file [rf4ce-gdp-internal.h](#).

8.14.1.48 #define CLIENT_NOTIFICATION_SUBTYPE_IDENTITY_PAYLOAD_LENGTH

Definition at line 161 of file [rf4ce-gdp-internal.h](#).

8.14.1.49 #define CLIENT_NOTIFICATION_SUBTYPE_IDENTITY_PAYLOAD_FLAGS_OFFSET

Definition at line 162 of file [rf4ce-gdp-internal.h](#).

8.14.1.50 #define CLIENT_NOTIFICATION_SUBTYPE_IDENTITY_PAYLOAD_TIME_OFFSET

Definition at line 163 of file [rf4ce-gdp-internal.h](#).

8.14.1.51 #define CLIENT_NOTIFICATION_SUBTYPE_REQUEST_POLL_NEGOTIATION_LENGTH

Definition at line 164 of file [rf4ce-gdp-internal.h](#).

8.14.1.52 #define CLIENT_NOTIFICATION_SUBTYPE_REQUEST_POLL_NEGOTIATION_PAYLOAD_LENGTH

Definition at line 165 of file [rf4ce-gdp-internal.h](#).

8.14.1.53 #define KEY_EXCHANGE_LENGTH

Definition at line 170 of file [rf4ce-gdp-internal.h](#).

8.14.1.54 #define KEY_EXCHANGE_SUBTYPE_OFFSET

Definition at line 171 of file [rf4ce-gdp-internal.h](#).

8.14.1.55 #define KEY_EXCHANGE_PAYLOAD_OFFSET

Definition at line 172 of file [rf4ce-gdp-internal.h](#).

8.14.1.56 #define KEY_EXCHANGE_SUBTYPE_CHALLENGE_LENGTH

Definition at line 173 of file [rf4ce-gdp-internal.h](#).

8.14.1.57 #define KEY_EXCHANGE_SUBTYPE_CHALLENGE_FLAGS_OFFSET

Definition at line 174 of file [rf4ce-gdp-internal.h](#).

8.14.1.58 #define KEY_EXCHANGE_SUBTYPE_CHALLENGE_RAND_A_OFFSET

Definition at line 175 of file [rf4ce-gdp-internal.h](#).

8.14.1.59 #define KEY_EXCHANGE_SUBTYPE_CHALLENGE_RESPONSE_LENGTH

Definition at line 176 of file [rf4ce-gdp-internal.h](#).

8.14.1.60 #define KEY_EXCHANGE_SUBTYPE_CHALLENGE_RESPONSE_FLAGS_OFFSET

Definition at line 177 of file [rf4ce-gdp-internal.h](#).

8.14.1.61 #define KEY_EXCHANGE_SUBTYPE_CHALLENGE_RESPONSE_RAND_B_OFFSET

Definition at line 178 of file [rf4ce-gdp-internal.h](#).

8.14.1.62 #define KEY_EXCHANGE_SUBTYPE_CHALLENGE_RESPONSE_TAG_B_OFFSET

Definition at line 179 of file [rf4ce-gdp-internal.h](#).

8.14.1.63 #define KEY_EXCHANGE_SUBTYPE_RESPONSE_LENGTH

Definition at line 180 of file [rf4ce-gdp-internal.h](#).

8.14.1.64 #define KEY_EXCHANGE_SUBTYPE_RESPONSE_TAG_A_OFFSET

Definition at line 181 of file [rf4ce-gdp-internal.h](#).

8.14.1.65 #define KEY_EXCHANGE_SUBTYPE_CONFIRM_LENGTH

Definition at line 182 of file [rf4ce-gdp-internal.h](#).

8.14.1.66 #define COMMAND_CODE_MAXIMUM

Definition at line 184 of file [rf4ce-gdp-internal.h](#).

8.14.1.67 #define GDP_CAPABILITIES_SUPPORT_EXTENDED_VALIDATION_BIT

Definition at line 187 of file [rf4ce-gdp-internal.h](#).

8.14.1.68 #define GDP_CAPABILITIES_SUPPORT_EXTENDED_VALIDATION_OFFSET

Definition at line 188 of file [rf4ce-gdp-internal.h](#).

8.14.1.69 #define GDP_CAPABILITIES_SUPPORT_POLL_SERVER_BIT

Definition at line 189 of file [rf4ce-gdp-internal.h](#).

8.14.1.70 #define GDP_CAPABILITIES_SUPPORT_POLL_SERVER_OFFSET

Definition at line 190 of file [rf4ce-gdp-internal.h](#).

8.14.1.71 #define GDP_CAPABILITIES_SUPPORT_POLL_CLIENT_BIT

Definition at line 191 of file [rf4ce-gdp-internal.h](#).

8.14.1.72 #define GDP_CAPABILITIES_SUPPORT_POLL_CLIENT_OFFSET

Definition at line 192 of file [rf4ce-gdp-internal.h](#).

8.14.1.73 #define GDP_CAPABILITIES_SUPPORT_IDENTIFICATION_SERVER_BIT

Definition at line 193 of file [rf4ce-gdp-internal.h](#).

8.14.1.74 #define GDP_CAPABILITIES_SUPPORT_IDENTIFICATION_SERVER_OFFSET

Definition at line 194 of file [rf4ce-gdp-internal.h](#).

8.14.1.75 #define GDP_CAPABILITIES_SUPPORT_IDENTIFICATION_CLIENT_BIT

Definition at line 195 of file [rf4ce-gdp-internal.h](#).

8.14.1.76 #define GDP_CAPABILITIES_SUPPORT_IDENTIFICATION_CLIENT_OFFSET

Definition at line 196 of file [rf4ce-gdp-internal.h](#).

8.14.1.77 #define GDP_CAPABILITIES_SUPPORT_ENHANCED_SECURITY_BIT

Definition at line 197 of file [rf4ce-gdp-internal.h](#).

8.14.1.78 #define GDP_CAPABILITIES_SUPPORT_ENHANCED_SECURITY_OFFSET

Definition at line 198 of file [rf4ce-gdp-internal.h](#).

8.14.1.79 #define GDP_CAPABILITIES_SUPPORT_SHARED_SECRET_OF_LOCAL_VENDOR_BIT

Definition at line 199 of file [rf4ce-gdp-internal.h](#).

8.14.1.80 #define GDP_CAPABILITIES_SUPPORT_SHARED_SECRET_OF_LOCAL_VENDOR_OFFSET

Definition at line 200 of file [rf4ce-gdp-internal.h](#).

8.14.1.81 #define GDP_CAPABILITIES_SUPPORT_SHARED_SECRET_OF_REMOTE_VENDOR_BIT

Definition at line 201 of file [rf4ce-gdp-internal.h](#).

8.14.1.82 #define GDP_CAPABILITIES_SUPPORT_SHARED_SECRET_OF_REMOTE_VENDOR_OFFSET

Definition at line 202 of file [rf4ce-gdp-internal.h](#).

8.14.1.83 #define GDP_STANDARD_SHARED_SECRET

Definition at line 206 of file [rf4ce-gdp-internal.h](#).

8.14.1.84 #define GDP_SHARED_SECRET_SIZE

Definition at line 209 of file [rf4ce-gdp-internal.h](#).

8.14.1.85 #define USER_STRING_APPLICATION_SPECIFIC_USER_STRING_OFFSET

Definition at line 212 of file [rf4ce-gdp-internal.h](#).

8.14.1.86 #define USER_STRING_APPLICATION_SPECIFIC_USER_STRING_LENGTH

Definition at line 213 of file [rf4ce-gdp-internal.h](#).

8.14.1.87 #define USER_STRING_NULL_BYTE_OFFSET

Definition at line 218 of file [rf4ce-gdp-internal.h](#).

8.14.1.88 #define USER_STRING_DISC_REQUEST_VENDOR_ID_FILTER_OFFSET

Definition at line 221 of file [rf4ce-gdp-internal.h](#).

8.14.1.89 #define USER_STRING_DISC_REQUEST_VENDOR_ID_FILTER_LENGTH

Definition at line 222 of file [rf4ce-gdp-internal.h](#).

8.14.1.90 #define USER_STRING_DISC_REQUEST_MIN_MAX_CLASS_FILTER_OFFSET

Definition at line 223 of file [rf4ce-gdp-internal.h](#).

8.14.1.91 #define USER_STRING_DISC_REQUEST_MIN_MAX_CLASS_FILTER_MIN_CLASS_NUM_MASK

Definition at line 224 of file [rf4ce-gdp-internal.h](#).

8.14.1.92 #define USER_STRING_DISC_REQUEST_MIN_MAX_CLASS_FILTER_MIN_CLASS_NUM_OFFSET

Definition at line 225 of file [rf4ce-gdp-internal.h](#).

8.14.1.93 #define USER_STRING_DISC_REQUEST_MIN_MAX_CLASS_FILTER_MAX_CLASS_NUM_MASK

Definition at line 226 of file [rf4ce-gdp-internal.h](#).

8.14.1.94 #define USER_STRING_DISC_REQUEST_MIN_MAX_CLASS_FILTER_MAX_CLASS_NUM_OFFSET

Definition at line 227 of file [rf4ce-gdp-internal.h](#).

8.14.1.95 #define USER_STRING_DISC_REQUEST_MIN_LQI_FILTER_OFFSET

Definition at line 228 of file [rf4ce-gdp-internal.h](#).

8.14.1.96 #define USER_STRING_DISC_REQUEST_RESERVED_BYTES_OFFSET

Definition at line 229 of file [rf4ce-gdp-internal.h](#).

8.14.1.97 #define USER_STRING_DISC_REQUEST_RESERVED_BYTES_LENGTH

Definition at line 230 of file [rf4ce-gdp-internal.h](#).

8.14.1.98 #define USER_STRING_DISC_RESPONSE_RESERVED_BYTES_OFFSET

Definition at line 233 of file [rf4ce-gdp-internal.h](#).

8.14.1.99 #define USER_STRING_DISC_RESPONSE_RESERVED_BYTES_LENGTH

Definition at line 234 of file [rf4ce-gdp-internal.h](#).

8.14.1.100 #define USER_STRING_DISC_RESPONSE_TERTIARY_CLASS_DESCRIPTOR_OFFSET

Definition at line 235 of file [rf4ce-gdp-internal.h](#).

8.14.1.101 #define USER_STRING_DISC_RESPONSE_SECONDARY_CLASS_DESCRIPTOR_OFFSET

Definition at line 236 of file [rf4ce-gdp-internal.h](#).

8.14.1.102 #define USER_STRING_DISC_RESPONSE_PRIMARY_CLASS_DESCRIPTOR_OFFSET

Definition at line 237 of file [rf4ce-gdp-internal.h](#).

8.14.1.103 #define USER_STRING_DISC_RESPONSE_DISCOVERY_LQI_THRESHOLD_OFFSET

Definition at line 238 of file [rf4ce-gdp-internal.h](#).

8.14.1.104 #define USER_STRING_PAIR_REQUEST_ADVANCED_BINDING_SUPPORT_OFFSET

Definition at line 239 of file [rf4ce-gdp-internal.h](#).

8.14.1.105 #define USER_STRING_PAIR_REQUEST_RESERVED_BYTES_OFFSET

Definition at line 240 of file [rf4ce-gdp-internal.h](#).

8.14.1.106 #define USER_STRING_PAIR_REQUEST_RESERVED_BYTES_LENGTH

Definition at line 241 of file [rf4ce-gdp-internal.h](#).

8.14.1.107 #define ADVANCED_BINDING_SUPPORT_FIELD_BINDING_PROXY_SUPPORTED_BIT

Definition at line 244 of file [rf4ce-gdp-internal.h](#).

8.14.1.108 #define CLASS_DESCRIPTOR_NUMBER_MASK

Definition at line 248 of file [rf4ce-gdp-internal.h](#).

8.14.1.109 #define CLASS_DESCRIPTOR_NUMBER_OFFSET

Definition at line 249 of file [rf4ce-gdp-internal.h](#).

8.14.1.110 #define CLASS_DESCRIPTOR_DUPLICATE_HANDLING_MASK

Definition at line 250 of file [rf4ce-gdp-internal.h](#).

8.14.1.111 #define CLASS_DESCRIPTOR_DUPLICATE_HANDLING_OFFSET

Definition at line 251 of file [rf4ce-gdp-internal.h](#).

8.14.1.112 `#define CLASS_DESCRIPTOR_RESERVED_MASK`

Definition at line 252 of file [rf4ce-gdp-internal.h](#).

8.14.1.113 `#define CLASS_NUMBER_PRE_COMMISSIONED`

Definition at line 254 of file [rf4ce-gdp-internal.h](#).

8.14.1.114 `#define CLASS_NUMBER_BUTTON_PRESS_INDICATION`

Definition at line 255 of file [rf4ce-gdp-internal.h](#).

8.14.1.115 `#define CLASS_NUMBER_DISCOVERABLE_ONLY`

Definition at line 257 of file [rf4ce-gdp-internal.h](#).

8.14.1.116 `#define CLASS_DESCRIPTOR_DUPLICATE_HANDLING_AS_IS`

Definition at line 260 of file [rf4ce-gdp-internal.h](#).

8.14.1.117 `#define CLASS_DESCRIPTOR_DUPLICATE_HANDLING_RECLASSIFY`

Definition at line 261 of file [rf4ce-gdp-internal.h](#).

8.14.1.118 `#define CLASS_DESCRIPTOR_DUPLICATE_HANDLING_ABORT`

Definition at line 262 of file [rf4ce-gdp-internal.h](#).

8.14.1.119 `#define CLASS_DESCRIPTOR_DUPLICATE_HANDLING_RESERVED`

Definition at line 263 of file [rf4ce-gdp-internal.h](#).

8.14.1.120 `#define AS_IS`

Definition at line 266 of file [rf4ce-gdp-internal.h](#).

8.14.1.121 `#define RECLASSIFY`

Definition at line 267 of file [rf4ce-gdp-internal.h](#).

8.14.1.122 `#define ABORT`

Definition at line 268 of file [rf4ce-gdp-internal.h](#).

8.14.1.123 `#define GDP_VERSION_NONE`

Definition at line 270 of file [rf4ce-gdp-internal.h](#).

8.14.1.124 `#define GDP_VERSION_1_X`

Definition at line 271 of file [rf4ce-gdp-internal.h](#).

8.14.1.125 `#define GDP_VERSION_2_0`

Definition at line 272 of file [rf4ce-gdp-internal.h](#).

8.14.1.126 `#define GDP_1_X_BASED_PROFILE_ID_LIST`

Definition at line 276 of file [rf4ce-gdp-internal.h](#).

8.14.1.127 `#define GDP_1_X_BASED_PROFILE_ID_LIST_LENGTH`

Definition at line 278 of file [rf4ce-gdp-internal.h](#).

8.14.1.128 `#define GDP_2_0_BASED_PROFILE_ID_LIST`

Definition at line 282 of file [rf4ce-gdp-internal.h](#).

8.14.1.129 `#define GDP_2_0_BASED_PROFILE_ID_LIST_LENGTH`

Definition at line 284 of file [rf4ce-gdp-internal.h](#).

8.14.1.130 `#define PUBLIC_STATE_MASK`

Definition at line 336 of file [rf4ce-gdp-internal.h](#).

8.14.1.131 `#define INTERNAL_STATE_MASK`

Definition at line 337 of file [rf4ce-gdp-internal.h](#).

8.14.1.132 `#define INTERNAL_STATE_OFFSET`

Definition at line 338 of file [rf4ce-gdp-internal.h](#).

8.14.1.133 `#define INTERNAL_STATE_NONE`

Definition at line 341 of file [rf4ce-gdp-internal.h](#).

8.14.1.134 #define INTERNAL_STATE_ORIGINATOR_GDP_CONFIG_PUSH_PENDING

Definition at line 343 of file [rf4ce-gdp-internal.h](#).

8.14.1.135 #define INTERNAL_STATE_ORIGINATOR_GDP_CONFIG_GET_PENDING

Definition at line 344 of file [rf4ce-gdp-internal.h](#).

8.14.1.136 #define INTERNAL_STATE_ORIGINATOR_GDP_CONFIG_PULL_PENDING

Definition at line 345 of file [rf4ce-gdp-internal.h](#).

8.14.1.137 #define INTERNAL_STATE_ORIGINATOR_GDP_CONFIG_COMPLETE_PENDING

Definition at line 346 of file [rf4ce-gdp-internal.h](#).

8.14.1.138 #define INTERNAL_STATE_ORIGINATOR_GDP_PROFILES_CONFIG

Definition at line 347 of file [rf4ce-gdp-internal.h](#).

8.14.1.139 #define INTERNAL_STATE_ORIGINATOR_GDP_VALIDATION

Definition at line 348 of file [rf4ce-gdp-internal.h](#).

8.14.1.140 #define INTERNAL_STATE_ORIGINATOR_GDP_KEY_EXCHANGE_BLACKOUT_PENDING

Definition at line 349 of file [rf4ce-gdp-internal.h](#).

8.14.1.141 #define INTERNAL_STATE_RECIPIENT_GDP_STACK_STATUS_NETWORK_UP_PENDING

Definition at line 351 of file [rf4ce-gdp-internal.h](#).

8.14.1.142 #define INTERNAL_STATE_RECIPIENT_GDP_RESTORE_PAIRING_ENTRY_PENDING

Definition at line 352 of file [rf4ce-gdp-internal.h](#).

8.14.1.143 #define INTERNAL_STATE_RECIPIENT_GDP_CONFIG_WAITING_FOR_PUSH

Definition at line 353 of file [rf4ce-gdp-internal.h](#).

8.14.1.144 #define INTERNAL_STATE_RECIPIENT_GDP_CONFIG_WAITING_FOR_GET

Definition at line 354 of file [rf4ce-gdp-internal.h](#).

8.14.1.145 #define INTERNAL_STATE_RECIPIENT_GDP_CONFIG_WAITING_FOR_PULL

Definition at line 355 of file [rf4ce-gdp-internal.h](#).

8.14.1.146 #define INTERNAL_STATE_RECIPIENT_GDP_CONFIG_WAITING_FOR_COMPLETE

Definition at line 356 of file [rf4ce-gdp-internal.h](#).

8.14.1.147 #define INTERNAL_STATE_RECIPIENT_GDP_PROFILES_CONFIG

Definition at line 357 of file [rf4ce-gdp-internal.h](#).

8.14.1.148 #define INTERNAL_STATE_RECIPIENT_GDP_VALIDATION

Definition at line 358 of file [rf4ce-gdp-internal.h](#).

8.14.1.149 #define INTERNAL_STATE_GDP_SECURITY_KEY_CHALLENGE_PENDING

Definition at line 360 of file [rf4ce-gdp-internal.h](#).

8.14.1.150 #define INTERNAL_STATE_GDP_SECURITY_KEY_CHALLENGE_RESPONSE_PENDING

Definition at line 361 of file [rf4ce-gdp-internal.h](#).

8.14.1.151 #define INTERNAL_STATE_GDP_SECURITY_KEY_RESPONSE_PENDING

Definition at line 362 of file [rf4ce-gdp-internal.h](#).

8.14.1.152 #define INTERNAL_STATE_GDP_POLL_CONFIG_CLIENT_PUSH_PENDING

Definition at line 364 of file [rf4ce-gdp-internal.h](#).

8.14.1.153 #define INTERNAL_STATE_GDP_POLL_CONFIG_CLIENT_PULL_PENDING

Definition at line 365 of file [rf4ce-gdp-internal.h](#).

8.14.1.154 #define INTERNAL_STATE_GDP_IDENTIFICATION_CLIENT_PUSH_PENDING

Definition at line 367 of file [rf4ce-gdp-internal.h](#).

8.14.1.155 #define INTERNAL_STATE_GDP_POLL_CLIENT_HEARTBEAT_PENDING

Definition at line 369 of file [rf4ce-gdp-internal.h](#).

8.14.1.156 `#define publicBindState()`

Definition at line 371 of file [rf4ce-gdp-internal.h](#).

8.14.1.157 `#define internalGdpState()`

Definition at line 372 of file [rf4ce-gdp-internal.h](#).

8.14.1.158 `#define setPublicState(state, init)`

Definition at line 375 of file [rf4ce-gdp-internal.h](#).

8.14.1.159 `#define setInternalState(state)`

Definition at line 389 of file [rf4ce-gdp-internal.h](#).

8.14.1.160 `#define isInternalStateBindingOriginator()`

Definition at line 398 of file [rf4ce-gdp-internal.h](#).

8.14.1.161 `#define isInternalStateBindingRecipient()`

Definition at line 402 of file [rf4ce-gdp-internal.h](#).

8.14.1.162 `#define isInternalStateSecurity()`

Definition at line 406 of file [rf4ce-gdp-internal.h](#).

8.14.1.163 `#define isInternalStatePollNegotiation()`

Definition at line 410 of file [rf4ce-gdp-internal.h](#).

8.14.1.164 `#define isInternalStateIdentification()`

Definition at line 414 of file [rf4ce-gdp-internal.h](#).

8.14.1.165 `#define CANDIDATE_INFO_ENTRY_IN_USE_BIT`

Definition at line 419 of file [rf4ce-gdp-internal.h](#).

8.14.1.166 `#define CANDIDATE_INFO_ENTRY_IN_USE_OFFSET`

Definition at line 420 of file [rf4ce-gdp-internal.h](#).

8.14.1.167 `#define CANDIDATE_INFO_PAIRING_ATTEMPTED_BIT`

Definition at line 421 of file [rf4ce-gdp-internal.h](#).

8.14.1.168 `#define CANDIDATE_INFO_PAIRING_ATTEMPTED_OFFSET`

Definition at line 422 of file [rf4ce-gdp-internal.h](#).

8.14.1.169 `#define CANDIDATE_INFO_PROXY_CANDIDATE_BIT`

Definition at line 423 of file [rf4ce-gdp-internal.h](#).

8.14.1.170 `#define CANDIDATE_INFO_PROXY_CANDIDATE_OFFSET`

Definition at line 424 of file [rf4ce-gdp-internal.h](#).

8.14.1.171 `#define PAIRING_ENTRY_BINDING_STATUS_MASK`

Definition at line 455 of file [rf4ce-gdp-internal.h](#).

8.14.1.172 `#define PAIRING_ENTRY_BINDING_STATUS_OFFSET`

Definition at line 456 of file [rf4ce-gdp-internal.h](#).

8.14.1.173 `#define PAIRING_ENTRY_BINDING_STATUS_NOT_BOUND`

Definition at line 457 of file [rf4ce-gdp-internal.h](#).

8.14.1.174 `#define PAIRING_ENTRY_BINDING_STATUS_BOUND_ORIGINATOR`

Definition at line 458 of file [rf4ce-gdp-internal.h](#).

8.14.1.175 `#define PAIRING_ENTRY_BINDING_STATUS_BOUND_RECIPIENT`

Definition at line 459 of file [rf4ce-gdp-internal.h](#).

8.14.1.176 `#define PAIRING_ENTRY_BINDING_COMPLETE_BIT`

Definition at line 460 of file [rf4ce-gdp-internal.h](#).

8.14.1.177 `#define PAIRING_ENTRY_BINDING_COMPLETE_OFFSET`

Definition at line 461 of file [rf4ce-gdp-internal.h](#).

8.14.1.178 #define PAIRING_ENTRY_POLLING_ACTIVE_BIT

Definition at line 462 of file [rf4ce-gdp-internal.h](#).

8.14.1.179 #define PAIRING_ENTRY_POLLING_ACTIVE_OFFSET

Definition at line 463 of file [rf4ce-gdp-internal.h](#).

8.14.1.180 #define PAIRING_ENTRY_IDENTIFICATION_ACTIVE_BIT

Definition at line 464 of file [rf4ce-gdp-internal.h](#).

8.14.1.181 #define PAIRING_ENTRY_IDENTIFICATION_ACTIVE_OFFSET

Definition at line 465 of file [rf4ce-gdp-internal.h](#).

8.14.1.182 #define PAIRING_ENTRY_REMOTE_NODE_SUPPORTS_ENHANCED_SECURITY_BIT

Definition at line 466 of file [rf4ce-gdp-internal.h](#).

8.14.1.183 #define PAIRING_ENTRY_REMOTE_NODE_SUPPORTS_ENHANCED_SECURITY_OFFSET

Definition at line 467 of file [rf4ce-gdp-internal.h](#).

8.14.1.184 #define PAIRING_ENTRY_REMOTE_NODE_SUPPORTS_IDENTIFICATION_BIT

Definition at line 468 of file [rf4ce-gdp-internal.h](#).

8.14.1.185 #define PAIRING_ENTRY_REMOTE_NODE_SUPPORTS_IDENTIFICATION_OFFSET

Definition at line 469 of file [rf4ce-gdp-internal.h](#).

8.14.1.186 #define PAIRING_ENTRY_REMOTE_NODE_SUPPORTS_POLLING_BIT

Definition at line 470 of file [rf4ce-gdp-internal.h](#).

8.14.1.187 #define PAIRING_ENTRY_REMOTE_NODE_SUPPORTS_POLLING_OFFSET

Definition at line 471 of file [rf4ce-gdp-internal.h](#).

8.14.1.188 #define emAfRf4ceGdpIsBusy()

Definition at line 485 of file [rf4ce-gdp-internal.h](#).

8.14.1.189 `#define GDP_SET_VALUE_BINDING_ORIGINATOR_PARAMETERS_BYTES_LENGTH`

Definition at line 577 of file [rf4ce-gdp-internal.h](#).

8.14.1.190 `#define GDP_SET_VALUE_BINDING_RECIPIENT_PARAMETERS_BYTES_LENGTH`

Definition at line 581 of file [rf4ce-gdp-internal.h](#).

8.14.1.191 `#define GDP_SET_VALUE_FLAG_LENGTH`

Definition at line 583 of file [rf4ce-gdp-internal.h](#).

8.14.1.192 `#define emAfRf4ceGdpOutgoingCommandsSetPendingFlag()`

Definition at line 597 of file [rf4ce-gdp-internal.h](#).

8.14.1.193 `#define emAfRf4ceGdpOutgoingCommandsClearPendingFlag()`

Definition at line 599 of file [rf4ce-gdp-internal.h](#).

8.14.1.194 `#define debugDiscoveryResponseDrop(reason)`

Definition at line 617 of file [rf4ce-gdp-internal.h](#).

8.14.1.195 `#define debugCandidateAdded(reason)`

Definition at line 618 of file [rf4ce-gdp-internal.h](#).

8.14.1.196 `#define debugScriptCheck(reason)`

Definition at line 619 of file [rf4ce-gdp-internal.h](#).

8.14.2 Function Documentation

8.14.2.1 `bool emAfIsProfileGdpBased(uint8_t profileId, uint8_t gdpCheckVersion)`

8.14.2.2 `bool emAfRf4ceIsProfileSupported(uint8_t profileId, const uint8_t * profileIdList, uint8_t profileIdListLength)`

8.14.2.3 `uint8_t emAfCheckDeviceTypeAndProfileIdMatch(uint8_t checkDeviceType, uint8_t * compareDevTypeList, uint8_t compareDevTypeListLength, uint8_t * checkProfileIdList, uint8_t checkProfileIdListLength, uint8_t * compareProfileIdList, uint8_t compareProfileIdListLength, uint8_t * matchingProfileIdList)`

8.14.2.4 `bool emAfRf4ceGdpMaybeStartNextProfileSpecificConfigurationProcedure(bool isOriginator, const uint8_t * remoteProfileIdList, uint8_t remoteProfileIdListLength)`

- 8.14.2.5 void emAfRf4ceGdpNotifyBindingCompleteToProfiles (EmberAfRf4ceGdpBindingStatus *status*, uint8_t *pairingIndex*, const uint8_t * *remoteProfileIdList*, uint8_t *remoteProfileIdListLength*)
- 8.14.2.6 EmberStatus emAfRf4ceGdpInitiateKeyExchangeInternal (uint8_t *pairingIndex*, bool *intCall*)
- 8.14.2.7 void emAfRf4ceGdpNoteProfileSpecificConfigurationStart (void)
- 8.14.2.8 void emAfGdpAddToProfileIdList (uint8_t * *srcProfileIdList*, uint8_t *srcProfileIdListLength*, EmberRf4ceApplicationInfo * *destAppInfo*, uint8_t *gdpVersion*)
- 8.14.2.9 uint8_t emAfRf4ceGdpGetGdpVersion (const uint8_t * *profileIdList*, uint8_t *profileIdListLength*)
- 8.14.2.10 void emAfRf4ceGdpRecipientInitCallback (void)
- 8.14.2.11 void emAfRf4ceGdpOriginatorStackStatusCallback (EmberStatus *status*)
- 8.14.2.12 void emAfRf4ceGdpRecipientStackStatusCallback (EmberStatus *status*)
- 8.14.2.13 void emAfRf4ceGdpUpdatePublicStatus (bool *init*)
- 8.14.2.14 uint8_t emAfRf4ceGdpGetPairingBindStatus (uint8_t *pairingIndex*)
- 8.14.2.15 void emAfRf4ceGdpSetPairingBindStatus (uint8_t *pairingIndex*, uint8_t *status*)
- 8.14.2.16 void emAfRf4ceGdpGetPairingKey (uint8_t *pairingIndex*, EmberKeyData * *key*)
- 8.14.2.17 void emAfRf4ceGdpSetPairingKey (uint8_t *pairingIndex*, EmberKeyData * *key*)
- 8.14.2.18 bool emAfRf4ceGdpSecurityGetRandomString (EmberAfRf4ceGdpRand * *rn*)
- 8.14.2.19 void emAfRf4ceGdpSecurityValidationCompleteCallback (uint8_t *pairingIndex*)
- 8.14.2.20 void emAfRf4ceGdpAttributesInitCallback (void)
- 8.14.2.21 void emAfRf4ceGdpIncomingGenericResponse (EmberAfRf4ceGdpResponseCode *responseCode*)
- 8.14.2.22 void emAfRf4ceGdpIncomingConfigurationComplete (EmberAfRf4ceGdpStatus *status*)
- 8.14.2.23 void emAfRf4ceGdpIncomingHeartbeat (EmberAfRf4ceGdpHeartbeatTrigger *trigger*)
- 8.14.2.24 void emAfRf4ceGdpIncomingGetAttributes (void)
- 8.14.2.25 void emAfRf4ceGdpIncomingGetAttributesResponse (void)
- 8.14.2.26 void emAfRf4ceGdpIncomingPushAttributes (void)
- 8.14.2.27 void emAfRf4ceGdpIncomingSetAttributes (void)
- 8.14.2.28 void emAfRf4ceGdpIncomingPullAttributes (void)

- 8.14.2.29 void emAfRf4ceGdpIncomingPullAttributesResponse (void)
- 8.14.2.30 void emAfRf4ceGdpIncomingCheckValidationRequest (uint8_t *control*)
- 8.14.2.31 void emAfRf4ceGdpIncomingCheckValidationResponse (EmberAfRf4ceGdpCheckValidationStatus *status*)
- 8.14.2.32 void emAfRf4ceGdpCheckValidationResponseSent (EmberStatus *status*)
- 8.14.2.33 void emAfRf4ceGdpHeartbeatSent (EmberStatus *status*)
- 8.14.2.34 void emAfRf4ceGdpIncomingKeyExchangeChallenge (EmberAfRf4ceGdpKeyExchangeFlags *flags*, const EmberAfRf4ceGdpRand * *randA*)
- 8.14.2.35 void emAfRf4ceGdpIncomingKeyExchangeChallengeResponse (EmberAfRf4ceGdpKeyExchangeFlags *flags*, const EmberAfRf4ceGdpRand * *randB*, const EmberAfRf4ceGdpTag * *tagB*)
- 8.14.2.36 void emAfRf4ceGdpIncomingKeyExchangeResponse (const EmberAfRf4ceGdpTag * *tagA*)
- 8.14.2.37 void emAfRf4ceGdpKeyExchangeResponseSent (EmberStatus *status*)
- 8.14.2.38 void emAfRf4ceGdpIncomingKeyExchangeConfirm (bool *secured*)
- 8.14.2.39 void emAfRf4ceGdpIncomingClientNotification (EmberAfRf4ceGdpClientNotificationSubtype *subType*, const uint8_t * *clientNotificationPayload*, uint8_t *clientNotificationPayloadLength*)
- 8.14.2.40 void emAfRf4ceZrcIncomingGenericResponse (EmberAfRf4ceGdpResponseCode *responseCode*)
- 8.14.2.41 void emAfRf4ceZrcIncomingConfigurationComplete (EmberAfRf4ceGdpStatus *status*)
- 8.14.2.42 void emAfRf4ceZrcIncomingGetAttributes (void)
- 8.14.2.43 void emAfRf4ceZrcIncomingGetAttributesResponse (void)
- 8.14.2.44 void emAfRf4ceZrcIncomingPushAttributes (void)
- 8.14.2.45 void emAfRf4ceZrcIncomingSetAttributes (void)
- 8.14.2.46 void emAfRf4ceZrcIncomingPullAttributes (void)
- 8.14.2.47 void emAfRf4ceZrcIncomingPullAttributesResponse (void)
- 8.14.2.48 void emAfRf4ceZrcIncomingClientNotification (EmberAfRf4ceGdpClientNotificationSubtype *subType*, const uint8_t * *clientNotificationPayload*, uint8_t *clientNotificationPayloadLength*)
- 8.14.2.49 bool emAfRf4ceGdpHasAttributeRecord (void)

- 8.14.2.50 `bool emAfRf4ceGdpAppendAttributIdentificationRecord (const EmberAfRf4ceGdpAttributeIdentificationRecord * record)`
- 8.14.2.51 `bool emAfRf4ceGdpFetchAttributIdentificationRecord (EmberAfRf4ceGdpAttributeIdentificationRecord * record)`
- 8.14.2.52 `bool emAfRf4ceGdpAppendAttributeStatusRecord (const EmberAfRf4ceGdpAttributeStatusRecord * record)`
- 8.14.2.53 `bool emAfRf4ceGdpFetchAttributeStatusRecord (EmberAfRf4ceGdpAttributeStatusRecord * record)`
- 8.14.2.54 `bool emAfRf4ceGdpAppendAttributeRecord (const EmberAfRf4ceGdpAttributeRecord * record)`
- 8.14.2.55 `bool emAfRf4ceGdpFetchAttributeRecord (EmberAfRf4ceGdpAttributeRecord * record)`
- 8.14.2.56 `void emAfRf4ceGdpResetFetchAttributeFinger (void)`
- 8.14.2.57 `void emAfRf4ceGdpStartAttributesCommand (EmberAfRf4ceGdpCommandCode commandCode)`
- 8.14.2.58 `EmberStatus emAfRf4ceGdpSendAttributesCommand (uint8_t pairingIndex, uint8_t profileId, uint16_t vendorId)`
- 8.14.2.59 `EmberStatus emAfRf4ceGdpSetDiscoveryResponseApplInfo (bool pushButton, uint8_t gapVersion)`
- 8.14.2.60 `EmberStatus emAfRf4ceGdpSetPairResponseApplInfo (const EmberRf4ceApplicationInfo * pairRequestApplInfo)`
- 8.14.2.61 `EmberStatus emAfRf4ceGdpSendProfileSpecificCommand (uint8_t pairingIndex, uint8_t profileId, uint16_t vendorId, EmberRf4ceTxOption txOptions, uint8_t commandId, uint8_t * commandPayload, uint8_t commandPayloadLength, uint8_t * messageTag)`
- 8.14.2.62 `EmberStatus emAfRf4ceGdpKeyExchangeChallenge (uint8_t pairingIndex, uint16_t vendorId, EmberAfRf4ceGdpKeyExchangeFlags flags, const EmberAfRf4ceGdpRand * randA)`
- 8.14.2.63 `EmberStatus emAfRf4ceGdpKeyExchangeChallengeResponse (uint8_t pairingIndex, uint16_t vendorId, EmberAfRf4ceGdpKeyExchangeFlags flags, const EmberAfRf4ceGdpRand * randB, const EmberAfRf4ceGdpTag * tagB)`
- 8.14.2.64 `EmberStatus emAfRf4ceGdpKeyExchangeResponse (uint8_t pairingIndex, uint16_t vendorId, const EmberAfRf4ceGdpTag * tagA)`
- 8.14.2.65 `EmberStatus emAfRf4ceGdpKeyExchangeConfirm (uint8_t pairingIndex, uint16_t vendorId)`
- 8.14.2.66 `void emAfRf4ceGdpSetPushButtonPendingReceivedFlag (bool set)`
- 8.14.2.67 `void emAfRf4ceGdpSetProxyBindingFlag (bool set)`
- 8.14.2.68 `void emAfGdpStartBlackoutTimer (uint8_t state)`

8.14.2.69 void emAfGdpStartCommandPendingTimer (uint8_t state, uint16_t timeMs)

8.14.3 Variable Documentation

- 8.14.3.1 const uint8_t emAfRf4ceGdpApplicationSpecificUserString[USER_STRING_APPLICATION_SPECIFIC_USER_STRING_LENGTH]
- 8.14.3.2 const uint8_t emAfGdp1xProfiles[]
- 8.14.3.3 const uint8_t emAfGdp20Profiles[]
- 8.14.3.4 uint8_t emAfTemporaryPairingIndex
- 8.14.3.5 uint8_t emAfCurrentProfileSpecificIndex
- 8.14.3.6 uint16_t emAfGdpState
- 8.14.3.7 uint32_t emberAfPluginRf4ceGdpCapabilities
- 8.14.3.8 EmberEventControl emberAfPluginRf4ceGdpPendingCommandEventControl
- 8.14.3.9 EmberEventControl emberAfPluginRf4ceGdpBlackoutTimeEventControl
- 8.14.3.10 EmberEventControl emberAfPluginRf4ceGdpValidationEventControl
- 8.14.3.11 EmAfBindingInfo emAfGdpPeerInfo
- 8.14.3.12 uint8_t emAfRf4ceGdpOutgoingCommandFrameControl

8.15 rf4ce-gdp-internal.h

```

00001 // Copyright 2014 Silicon Laboratories, Inc.
00002
00003 #define NWK_DISCOVERY_REPEATITION_INTERVAL_MS 500
00004 #define NWK_MAX_DISCOVERY_REPEATITIONS 2
00005 #define NWK_MAX_REPORTED_NODE_DESCRIPTOROS 0xFF
00006
00007 // TODO: according to the specs, this parameter is implementation specific,
00008 // therefore it will eventually become one of the plugin options.
00009 #define NWK_DISCOVERY_LQI_THRESHOLD 0x01
00010
00011 // TODO: according to the specs, this value defaults to 3, however it can still
00012 // be configured by the application, therefore it will eventually become one of
00013 // the plugin options.
00014 #define APL_MAX_PAIRING_CANDIDATES 3
00015
00016 // The length of time after completing the pairing procedure or a configuration
00017 // phase of a profile during which no packets shall be sent to allow the remote
00018 // nodes to perform internal housekeeping tasks.
00019 #define APLC_CONFIG_BLACKOUT_TIME_MS 100
00020
00021 #define BLACKOUT_TIME_DELTA_MS 5
00022
00023 // At the originator the blackout time is increased by a delta value, to ensure
00024 // that the recipient is already on listening for packets.
00025 #define BLACKOUT_TIME_ORIGINATOR_MS (APLC_CONFIG_BLACKOUT_TIME_MS +
00026     BLACKOUT_TIME_DELTA_MS)
00026 // At the recipient the blackout time is reduced by a delta value, to ensure
00027 // that we are listening for incoming packets from the originator.
00028 #define BLACKOUT_TIME_RECIPIENT_MS (APLC_CONFIG_BLACKOUT_TIME_MS -
00029     BLACKOUT_TIME_DELTA_MS)
00030 // The duration of the binding window. On a Binding Originator, section 7.2.1

```

```

00031 // describes how this constant is used. On the Binding Recipient, section
00032 // 7.2.3.2 describes how this constant is used.
00033 #define APLC_BIND_WINDOW_DURATION_MS (30 * MILLISECOND_TICKS_PER_SECOND)
00034
00035 // The maximum allowed value for the aplMaxAutoCheckValidationPeriod attribute.
00036 #define APLC_MAX_AUTO_CHECK_VALIDATION_PERIOD_MS (10 *
    MILLISECOND_TICKS_PER_SECOND)
00037
00038 // The maximum time the Binding Recipient shall wait to receive a command frame
00039 // from a Binding Originator during its configuration phase.
00040 #define APLC_MAX_CONFIG_WAIT_TIME_MS 100
00041
00042 // The maximal time the validation can take in normal validation mode (see
00043 // section 7.2.7).
00044 #define APLC_MAX_NORMAL_VALIDATION_DURATION_MS      25000
00045
00046 // The maximal time the validation can take in extended validation mode (see
00047 // section 7.2.7).
00048 #define APLC_MAX_EXTENDED_VALIDATION_DURATION_MS   65000
00049
00050 // The maximum allowed value to configure the polling timeout in the
00051 // ap1PollConfiguration attribute.
00052 #define APLC_MAX_POLLING_TIMEOUT_MS 100
00053
00054 // The maximum time a node shall leave its receiver on in order to receive data
00055 // indicated via the data pending subfield of the frame control field of an
00056 // incoming frame.
00057 #define APLC_MAX_RX_ON_WAIT_TIME_MS 100
00058
00059 // The maximum time a node shall wait for a response command frame following a
00060 // request command frame.
00061 #define APLC_MAX_RESPONSE_WAIT_TIME_MS 100
00062
00063 // The minimum value of the KeyExchangeTransferCount parameter passed to the
00064 // pair request primitive during the validation based pairing procedure.
00065 #define APLC_MIN_KEY_EXCHANGE_TRANSFER_COUNT 3
00066
00067 // GDP header
00068 // - Frame control (1 byte)
00069 #define GDP_HEADER_LENGTH                      1
00070 #define GDP_HEADER_FRAME_CONTROL_OFFSET        0
00071 #define GDP_HEADER_FRAME_CONTROL_COMMAND_CODE_MASK 0x0F
00072 #define GDP_HEADER_FRAME_CONTROL_GDP_COMMAND_FRAME_MASK 0x40
00073 #define GDP_HEADER_FRAME_CONTROL_DATA_PENDING_MASK 0x80
00074 #define GDP_PAYLOAD_OFFSET                     1
00075
00076 // Generic Response
00077 // - Response code (1 byte)
00078 #define GENERIC_RESPONSE_LENGTH           (GDP_HEADER_LENGTH + 1)
00079 #define GENERIC_RESPONSE_RESPONSE_CODE_OFFSET (GDP_HEADER_LENGTH)
00080
00081 // Configuration Complete
00082 // - Status (1 byte)
00083 #define CONFIGURATION_COMPLETE_LENGTH       (GDP_HEADER_LENGTH + 1)
00084 #define CONFIGURATION_COMPLETE_STATUS_OFFSET (GDP_HEADER_LENGTH)
00085
00086 // Heartbeat
00087 // - Trigger (1 byte)
00088 #define HEARTBEAT_LENGTH                  (GDP_HEADER_LENGTH + 1)
00089 #define HEARTBEAT_TRIGGER_OFFSET          (GDP_HEADER_LENGTH)
00090
00091 // Get Attributes
00092 // - Attribute identification record (1/3 bytes)
00093 //   - Attribute identifier (1 byte)
00094 //   - Entry identifier (0/2 bytes)
00095 // ...
00096 #define GET_ATTRIBUTES_LENGTH          (GDP_HEADER_LENGTH + 1)
00097
00098 // Get Attributes Response
00099 // - Attribute status record (n bytes)
00100 //   - Attribute identifier (1 byte)
00101 //   - Entry identifier (0/2 bytes)
00102 //   - Attribute status (1 byte)
00103 //   - Attribute length (0/1 byte)
00104 //   - Attribute value (n bytes)
00105 // ...
00106 #define GET_ATTRIBUTES_RESPONSE_LENGTH (GDP_HEADER_LENGTH + 2)
00107
00108 // Push Attributes
00109 // - Attribute record (n bytes)

```

```

00110 // - Attribute identifier (1 byte)
00111 // - Entry identifier (0/2 bytes)
00112 // - Attribute length (1 byte)
00113 // - Attribute value (n bytes)
00114 // ...
00115 #define PUSH_ATTRIBUTES_LENGTH (GDP_HEADER_LENGTH + 2)
00116
00117 // Set Attributes
00118 // - Attribute record (n bytes)
00119 // - Attribute identifier (1 byte)
00120 // - Entry identifier (0/2 bytes)
00121 // - Attribute length (1 byte)
00122 // - Attribute value (n bytes)
00123 // ...
00124 #define SET_ATTRIBUTES_LENGTH (GDP_HEADER_LENGTH + 2)
00125
00126 // Pull Attributes
00127 // - Attribute identification record (1/3 bytes)
00128 // - Attribute identifier (1 byte)
00129 // - Entry identifier (0/2 bytes)
00130 // ...
00131 #define PULL_ATTRIBUTES_LENGTH (GDP_HEADER_LENGTH + 1)
00132
00133 // Pull Attributes Response
00134 // - Attribute record (n bytes)
00135 // - Attribute identifier (1 byte)
00136 // - Entry identifier (0/2 bytes)
00137 // - Attribute status (1 byte)
00138 // - Attribute length (0/1 byte)
00139 // - Attribute value (n bytes)
00140 // ...
00141 #define PULL_ATTRIBUTES_RESPONSE_LENGTH (GDP_HEADER_LENGTH + 2)
00142
00143 // Check Validation
00144 // - Check validation subtype (1 byte)
00145 // - Check validation payload (n bytes)
00146 #define CHECK_VALIDATION_LENGTH (GDP_HEADER_LENGTH + 1)
00147 #define CHECK_VALIDATION_SUBTYPE_OFFSET (GDP_HEADER_LENGTH)
00148 #define CHECK_VALIDATION_PAYLOAD_OFFSET (GDP_HEADER_LENGTH + 1)
00149 #define CHECK_VALIDATION_SUBTYPE_REQUEST_LENGTH (GDP_HEADER_LENGTH + 2)
00150 #define CHECK_VALIDATION_SUBTYPE_REQUEST_CONTROL_OFFSET (GDP_HEADER_LENGTH + 1)
00151 #define CHECK_VALIDATION_SUBTYPE_RESPONSE_LENGTH (GDP_HEADER_LENGTH + 2)
00152 #define CHECK_VALIDATION_SUBTYPE_RESPONSE_STATUS_OFFSET (GDP_HEADER_LENGTH + 1)
00153
00154 // Client Notification
00155 // - Client notification subtype (1 byte)
00156 // - Client notification payload (n bytes)
00157 #define CLIENT_NOTIFICATION_LENGTH (GDP_HEADER_LENGTH + 1)
00158 #define CLIENT_NOTIFICATION_SUBTYPE_OFFSET (GDP_HEADER_LENGTH)
00159 #define CLIENT_NOTIFICATION_PAYLOAD_OFFSET (GDP_HEADER_LENGTH + 1)
00160 #define CLIENT_NOTIFICATION_SUBTYPE_IDENTIFY_LENGTH (GDP_HEADER_LENGTH + 4)
00161 #define CLIENT_NOTIFICATION_SUBTYPE_IDENTIFY_PAYLOAD_LENGTH 3
00162 #define CLIENT_NOTIFICATION_SUBTYPE_IDENTIFY_PAYLOAD_FLAGS_OFFSET 0
00163 #define CLIENT_NOTIFICATION_SUBTYPE_IDENTIFY_PAYLOD_TIME_OFFSET 1
00164 #define CLIENT_NOTIFICATION_SUBTYPE_REQUEST_POLL_NEGOTIATION_LENGTH (GDP_HEADER_LENGTH + 1)
00165 #define CLIENT_NOTIFICATION_SUBTYPE_REQUEST_POLL_NEGOTIATION_PAYLOAD_LENGTH 0
00166
00167 // Key Exchange
00168 // - Key exchange subtype (1 byte)
00169 // - Key exchange payload (n bytes)
00170 #define KEY_EXCHANGE_LENGTH (GDP_HEADER_LENGTH + 1)
00171 #define KEY_EXCHANGE_SUBTYPE_OFFSET (GDP_HEADER_LENGTH)
00172 #define KEY_EXCHANGE_PAYLOAD_OFFSET (GDP_HEADER_LENGTH + 1)
00173 #define KEY_EXCHANGE_SUBTYPE_CHALLENGE_LENGTH (GDP_HEADER_LENGTH + 11)
00174 #define KEY_EXCHANGE_SUBTYPE_CHALLENGE_FLAGS_OFFSET (GDP_HEADER_LENGTH + 1)
00175 #define KEY_EXCHANGE_SUBTYPE_CHALLENGE RAND A _OFFSET (GDP_HEADER_LENGTH + 3)
00176 #define KEY_EXCHANGE_SUBTYPE_CHALLENGE_RESPONSE_LENGTH (GDP_HEADER_LENGTH + 15)

```

```

00177 #define KEY_EXCHANGE_SUBTYPE_CHALLENGE_RESPONSE_FLAGS_OFFSET
        (GDP_HEADER_LENGTH + 1)
00178 #define KEY_EXCHANGE_SUBTYPE_CHALLENGE_RESPONSE RAND_B_OFFSET
        (GDP_HEADER_LENGTH + 3)
00179 #define KEY_EXCHANGE_SUBTYPE_CHALLENGE_RESPONSE_TAG_B_OFFSET
        (GDP_HEADER_LENGTH + 11)
00180 #define KEY_EXCHANGE_SUBTYPE_RESPONSE_LENGTH
        (GDP_HEADER_LENGTH + 5)
00181 #define KEY_EXCHANGE_SUBTYPE_RESPONSE_TAG_A_OFFSET
        (GDP_HEADER_LENGTH + 1)
00182 #define KEY_EXCHANGE_SUBTYPE_CONFIRM_LENGTH
        (GDP_HEADER_LENGTH + 1)
00183
00184 #define COMMAND_CODE_MAXIMUM EMBER_AF_RF4CE_GDP_COMMAND_KEY_EXCHANGE
00185
00186 // GDP capabilities
00187 #define GDP_CAPABILITIES_SUPPORT_EXTENDED_VALIDATION_BIT
        0x00000001
00188 #define GDP_CAPABILITIES_SUPPORT_EXTENDED_VALIDATION_OFFSET      0
00189 #define GDP_CAPABILITIES_SUPPORT_POLL_SERVER_BIT
        0x00000002
00190 #define GDP_CAPABILITIES_SUPPORT_POLL_SERVER_OFFSET                1
00191 #define GDP_CAPABILITIES_SUPPORT_POLL_CLIENT_BIT
        0x00000004
00192 #define GDP_CAPABILITIES_SUPPORT_POLL_CLIENT_OFFSET               2
00193 #define GDP_CAPABILITIES_SUPPORT_IDENTIFICATION_SERVER_BIT
        0x00000008
00194 #define GDP_CAPABILITIES_SUPPORT_IDENTIFICATION_SERVER_OFFSET   3
00195 #define GDP_CAPABILITIES_SUPPORT_IDENTIFICATION_CLIENT_BIT
        0x00000010
00196 #define GDP_CAPABILITIES_SUPPORT_IDENTIFICATION_CLIENT_OFFSET  4
00197 #define GDP_CAPABILITIES_SUPPORT_ENHANCED_SECURITY_BIT
        0x00000020
00198 #define GDP_CAPABILITIES_SUPPORT_ENHANCED_SECURITY_OFFSET         5
00199 #define GDP_CAPABILITIES_SUPPORT_SHARED_SECRET_OF_LOCAL_VENDOR_BIT
        0x00000040
00200 #define GDP_CAPABILITIES_SUPPORT_SHARED_SECRET_OF_LOCAL_VENDOR_OFFSET 6
00201 #define GDP_CAPABILITIES_SUPPORT_SHARED_SECRET_OF_REMOTE_VENDOR_BIT
        0x00000080
00202 #define GDP_CAPABILITIES_SUPPORT_SHARED_SECRET_OF_REMOTE_VENDOR_OFFSET 7
00203
00204 // Bits 8-31 are reserved
00205
00206 #define GDP_STANDARD_SHARED_SECRET {0x00, 0x11, 0x22, 0x33, 0x44, 0x55, 0x66,
00207     \
00208             0x77, 0x88, 0x99, 0xAA, 0xBB, 0xCC, 0xDD,
00209     \
00210             0xEE, 0xFF}
00209 #define GDP_SHARED_SECRET_SIZE      16
00210
00211 // Application specific user string.
00212 #define USER_STRING_APPLICATION_SPECIFIC_USER_STRING_OFFSET      0
00213 #define USER_STRING_APPLICATION_SPECIFIC_USER_STRING_LENGTH    8
00214
00215 extern const uint8_t emAfRf4ceGdpApplicationSpecificUserString
        [USER_STRING_APPLICATION_SPECIFIC_USER_STRING_LENGTH
        ];
00216
00217 // Null byte delimiter
00218 #define USER_STRING_NULL_BYTE_OFFSET                         8
00219
00220 // Discovery request user string bytes.
00221 #define USER_STRING_DISC_REQUEST_VENDOR_ID_FILTER_OFFSET      9
00222 #define USER_STRING_DISC_REQUEST_VENDOR_ID_FILTER_LENGTH    2
00223 #define USER_STRING_DISC_REQUEST_MIN_MAX_CLASS_FILTER_OFFSET 11
00224 #define USER_STRING_DISC_REQUEST_MIN_MAX_CLASS_FILTER_MIN_CLASS_NUM_MASK
        0x0F
00225 #define USER_STRING_DISC_REQUEST_MIN_MAX_CLASS_FILTER_MIN_CLASS_NUM_OFFSET 0
00226 #define USER_STRING_DISC_REQUEST_MIN_MAX_CLASS_FILTER_MAX_CLASS_NUM_MASK
        0xF0
00227 #define USER_STRING_DISC_REQUEST_MIN_MAX_CLASS_FILTER_MAX_CLASS_NUM_OFFSET 4
00228 #define USER_STRING_DISC_REQUEST_MIN_LQI_FILTER_OFFSET          12
00229 #define USER_STRING_DISC_REQUEST_RESERVED_BYTES_OFFSET        13
00230 #define USER_STRING_DISC_REQUEST_RESERVED_BYTES_LENGTH        2
00231
00232 // Discovery response user string bytes.
00233 #define USER_STRING_DISC_RESPONSE_RESERVED_BYTES_OFFSET        9
00234 #define USER_STRING_DISC_RESPONSE_RESERVED_BYTES_LENGTH        2
00235 #define USER_STRING_DISC_RESPONSE_TERTIARY_CLASS_DESCRIPTOR_OFFSET
        11
00236 #define USER_STRING_DISC_RESPONSE_SECONDARY_CLASS_DESCRIPTOR_OFFSET
        12

```



```

00302                                     uint8_t *matchingProfileIdList);
00303
00304 bool emAfRf4ceGdpMaybeStartNextProfileSpecificConfigurationProcedure
00305     (bool isOriginator,
00306      uint8_t *remoteProfileIdList,
00307      uint8_t remoteProfileIdListLength);
00308 void emAfRf4ceGdpNotifyBindingCompleteToProfiles
00309     (EmberAfRf4ceGdpBindingStatus status,
00310      uint8_t pairingIndex,
00311      const uint8_t *
00312      remoteProfileIdList,
00313      uint8_t remoteProfileIdListLength);
00314 EmberStatus emAfRf4ceGdpInitiateKeyExchangeInternal
00315     (uint8_t pairingIndex,
00316      bool intCall);
00317 void emAfRf4ceGdpNoteProfileSpecificConfigurationStart
00318     (void);
00319 void emAfGdpAddToProfileIdList(uint8_t *
00320     srcProfileIdList,
00321     uint8_t srcProfileIdListLength,
00322     EmberRf4ceApplicationInfo *destAppInfo,
00323     uint8_t gdpVersion);
00324 uint8_t emAfRf4ceGdpGetGdpVersion(const uint8_t *
00325     profileIdList,
00326     uint8_t profileIdListLength);
00327 extern uint8_t emAfTemporaryPairingIndex;
00328 extern uint8_t emAfCurrentProfileSpecificIndex;
00329 extern uint16_t emAfGdpState;
00330 extern uint32_t emberAfPluginRf4ceGdpCapabilities
00331 ;
00332 extern EmberEventControl emberAfPluginRf4ceGdpPendingCommandEventControl
00333 ;
00334 extern EmberEventControl emberAfPluginRf4ceGdpBlackoutTimeEventControl
00335 ;
00336 extern EmberEventControl emberAfPluginRf4ceGdpValidationEventControl
00337 ;
00338 // We use the emAfGdpState to store both the public (dormant, not-bound,
00339 // binding and bound) state and the internal states.
00340 #define PUBLIC_STATE_MASK 0x03
00341 #define INTERNAL_STATE_MASK 0xFFFF
00342 #define INTERNAL_STATE_OFFSET 2
00343
00344 // Internal states
00345 #define INTERNAL_STATE_NONE (0x00 << INTERNAL_STATE_OFFSET)
00346 // Originator binding states
00347 #define INTERNAL_STATE_ORIGINATOR_GDP_CONFIG_PUSH_PENDING (0x01 << INTERNAL_STATE_OFFSET)
00348 #define INTERNAL_STATE_ORIGINATOR_GDP_CONFIG_GET_PENDING (0x02 << INTERNAL_STATE_OFFSET)
00349 #define INTERNAL_STATE_ORIGINATOR_GDP_CONFIG_PULL_PENDING (0x03 << INTERNAL_STATE_OFFSET)
00350 #define INTERNAL_STATE_ORIGINATOR_GDP_CONFIG_COMPLETE_PENDING (0x04 << INTERNAL_STATE_OFFSET)
00351 #define INTERNAL_STATE_ORIGINATOR_GDP_PROFILES_CONFIG (0x05 << INTERNAL_STATE_OFFSET)
00352 #define INTERNAL_STATE_ORIGINATOR_GDP_VALIDATION (0x06 << INTERNAL_STATE_OFFSET)
00353 #define INTERNAL_STATE_ORIGINATOR_GDP_KEY_EXCHANGE_BLACKOUT_PENDING (0x07 << INTERNAL_STATE_OFFSET)
00354 // Recipient binding states
00355 #define INTERNAL_STATE_RECIPIENT_GDP_STACK_STATUS_NETWORK_UP_PENDING (0x08 << INTERNAL_STATE_OFFSET)
00356 #define INTERNAL_STATE_RECIPIENT_GDP_RESTORE_PAIRING_ENTRY_PENDING (0x09 << INTERNAL_STATE_OFFSET)
00357 #define INTERNAL_STATE_RECIPIENT_GDP_CONFIG_WAITING_FOR_PUSH (0x0A << INTERNAL_STATE_OFFSET)
00358 #define INTERNAL_STATE_RECIPIENT_GDP_CONFIG_WAITING_FOR_GET (0x0B << INTERNAL_STATE_OFFSET)
00359 #define INTERNAL_STATE_RECIPIENT_GDP_CONFIG_WAITING_FOR_PULL (0x0C <<

```

```

INTERNAL_STATE_OFFSET)
00356 #define INTERNAL_STATE_RECIPIENT_GDP_CONFIG_WAITING_FOR_COMPLETE (0x0D <<
INTERNAL_STATE_OFFSET)
00357 #define INTERNAL_STATE_RECIPIENT_GDP_PROFILES_CONFIG (0x0E <<
INTERNAL_STATE_OFFSET)
00358 #define INTERNAL_STATE_RECIPIENT_GDP_VALIDATION (0x0F <<
INTERNAL_STATE_OFFSET)
00359 // Security key exchange procedure states
00360 #define INTERNAL_STATE_GDP_SECURITY_KEY_CHALLENGE_PENDING (0x10 <<
INTERNAL_STATE_OFFSET)
00361 #define INTERNAL_STATE_GDP_SECURITY_KEY_CHALLENGE_RESPONSE_PENDING (0x11 <<
INTERNAL_STATE_OFFSET)
00362 #define INTERNAL_STATE_GDP_SECURITY_KEY_RESPONSE_PENDING (0x12 <<
INTERNAL_STATE_OFFSET)
00363 // Poll negotiation procedure states
00364 #define INTERNAL_STATE_GDP_POLL_CONFIG_CLIENT_PUSH_PENDING (0x13 <<
INTERNAL_STATE_OFFSET)
00365 #define INTERNAL_STATE_GDP_POLL_CONFIG_CLIENT_PULL_PENDING (0x14 <<
INTERNAL_STATE_OFFSET)
00366 // Identification procedure states
00367 #define INTERNAL_STATE_GDP_IDENTIFICATION_CLIENT_PUSH_PENDING (0x15 <<
INTERNAL_STATE_OFFSET)
00368 // Poll states
00369 #define INTERNAL_STATE_GDP_POLL_CLIENT_HEARTBEAT_PENDING (0x16 <<
INTERNAL_STATE_OFFSET)
00370
00371 #define publicBindState() (emAfGdpState & PUBLIC_STATE_MASK)
00372 #define internalGdpState() (emAfGdpState & INTERNAL_STATE_MASK)
00373
00374 // Setting the public state clears the internal state (implicitly set to NONE).
00375 #define setPublicState(state, init)
00376     \
00377     do {
00378         \
00379         emAfGdpState = (state);
00380         \
00381         if (!init) {
00382             \
00383             emberAfRf4ceRxEnable(EMBER_AF_RF4CE_PROFILE_GENERIC_DEVICE, false);
00384         }
00385     \
00386     } while(0)
00387
00388 // For any state other than NONE we keep the receiver ON.
00389 // The "STACK_STATUS_NETWORK_UP_PENDING" is a special state that is set in the
00390 // init function at the recipient. Since we can not control the order of the
00391 // plugin init functions calls, we might end up calling the RxEnable before the
00392 // profile plugin has been initialized. For this reason, we don't call the
00393 // RxEnable() in this state.
00394 #define setInternalState(state)
00395     \
00396     do {
00397         \
00398         emAfGdpState = ((emAfGdpState & PUBLIC_STATE_MASK) | (state));
00399         \
00400         if ((state) != INTERNAL_STATE_RECIPIENT_GDP_STACK_STATUS_NETWORK_UP_PENDING) { \
00401             \
00402             emberAfRf4ceRxEnable(EMBER_AF_RF4CE_PROFILE_GENERIC_DEVICE,
00403                                     ((state) != INTERNAL_STATE_NONE));
00404         }
00405     \
00406     } while(0)
00407
00408 #define isInternalStateBindingOriginator()
00409     \
00410     (internalGdpState() >= INTERNAL_STATE_ORIGINATOR_GDP_CONFIG_PUSH_PENDING
00411     \
00412     && internalGdpState() <=
INTERNAL_STATE_ORIGINATOR_GDP_KEY_EXCHANGE_BLACKOUT_PENDING)
00413
00414 #define isInternalStateBindingRecipient()
00415     \
00416     (internalGdpState() >=
INTERNAL_STATE_RECIPIENT_GDP_STACK_STATUS_NETWORK_UP_PENDING \
00417     && internalGdpState() <= INTERNAL_STATE_RECIPIENT_GDP_VALIDATION)
00418
00419 #define isInternalStateSecurity()

```

```

00407 \
00408   (internalGdpState() >= INTERNAL_STATE_GDP_SECURITY_KEY_CHALLENGE_PENDING
00409   && internalGdpState() <= INTERNAL_STATE_GDP_SECURITY_KEY_RESPONSE_PENDING)
00410 #define isInternalStatePollNegotiation()
00411 \
00412   (internalGdpState() >= INTERNAL_STATE_GDP_POLL_CONFIG_CLIENT_PUSH_PENDING
00413   && internalGdpState() <= INTERNAL_STATE_GDP_POLL_CONFIG_CLIENT_PULL_PENDING)
00414 #define isInternalStateIdentification()
00415 \
00416   (internalGdpState() == INTERNAL_STATE_GDP_IDENTIFICATION_CLIENT_PUSH_PENDING)
00417 // Pairing candidate info byte.
00418
00419 #define CANDIDATE_INFO_ENTRY_IN_USE_BIT          0x01
00420 #define CANDIDATE_INFO_ENTRY_IN_USE_OFFSET        0
00421 #define CANDIDATE_INFO_PAIRING_ATTEMPTED_BIT      0x02
00422 #define CANDIDATE_INFO_PAIRING_ATTEMPTED_OFFSET   1
00423 #define CANDIDATE_INFO_PROXY_CANDIDATE_BIT        0x04
00424 #define CANDIDATE_INFO_PROXY_CANDIDATE_OFFSET     2
00425
00426 typedef struct {
00427   EmberEUI64 ieeeAddr;
00428   EmberPanId panId;
00429   uint8_t supportedProfiles[EMBER_RF4CE_APPLICATION_PROFILE_ID_LIST_MAX_LENGTH];
00430   uint8_t supportedProfilesLength;
00431   uint8_t channel;
00432   uint8_t primaryClassDescriptor;
00433   uint8_t secondaryClassDescriptor;
00434   uint8_t tertiaryClassDescriptor;
00435   uint8_t rxLqi;
00436   uint8_t info;
00437 } EmAfGdpPairingCandidate;
00438
00439 typedef struct {
00440   EmberEUI64 srcIeeeAddr;
00441   uint8_t nodeCapabilities;
00442   EmberRf4ceVendorInfo vendorInfo;
00443   EmberRf4ceApplicationInfo appInfo;
00444   uint8_t searchDevType;
00445 } EmAfDiscoveryOrPairrequestData;
00446
00447 typedef struct {
00448   uint8_t localConfigurationStatus;
00449   uint8_t candidateIndex;
00450 } EmAfBindingInfo;
00451
00452 extern EmAfBindingInfo emAfGdpPeerInfo;
00453
00454 // Pairing table entry bind status
00455 #define PAIRING_ENTRY_BINDING_STATUS_MASK          0x03
00456 #define PAIRING_ENTRY_BINDING_STATUS_OFFSET        0
00457 #define PAIRING_ENTRY_BINDING_STATUS_NOT_BOUND    0x00
00458 #define PAIRING_ENTRY_BINDING_STATUS_BOUND_ORIGINATOR 0x01
00459 #define PAIRING_ENTRY_BINDING_STATUS_BOUND_RECIPIENT 0x02
00460 #define PAIRING_ENTRY_BINDING_COMPLETE_BIT         0x04
00461 #define PAIRING_ENTRY_BINDING_COMPLETE_OFFSET      2
00462 #define PAIRING_ENTRY_POLLING_ACTIVE_BIT          0x08
00463 #define PAIRING_ENTRY_POLLING_ACTIVE_OFFSET        3
00464 #define PAIRING_ENTRY_IDENTIFICATION_ACTIVE_BIT   0x10
00465 #define PAIRING_ENTRY_IDENTIFICATION_ACTIVE_OFFSET 4
00466 #define PAIRING_ENTRY_REMOTE_NODE_SUPPORTS_ENHANCED_SECURITY_BIT 0x20
00467 #define PAIRING_ENTRY_REMOTE_NODE_SUPPORTS_ENHANCED_SECURITY_OFFSET 5
00468 #define PAIRING_ENTRY_REMOTE_NODE_SUPPORTS_IDENTIFICATION_BIT 0x40
00469 #define PAIRING_ENTRY_REMOTE_NODE_SUPPORTS_IDENTIFICATION_OFFSET 6
00470 #define PAIRING_ENTRY_REMOTE_NODE_SUPPORTS_POLLING_BIT 0x80
00471 #define PAIRING_ENTRY_REMOTE_NODE_SUPPORTS_POLLING_OFFSET 7
00472
00473 void emAfRf4ceGdpRecipientInitCallback(void);
00474 void emAfRf4ceGdpOriginatorStackStatusCallback
  (EmberStatus status);
00475 void emAfRf4ceGdpRecipientStackStatusCallback
  (EmberStatus status);
00476
00477 void emAfRf4ceGdpUpdatePublicStatus(bool init);
00478

```

```

00479 uint8_t emAfRf4ceGdpGetPairingBindStatus(
00480     uint8_t pairingIndex);
00481 void emAfRf4ceGdpSetPairingBindStatus(uint8_t
00482     pairingIndex, uint8_t status);
00483 void emAfRf4ceGdpGetPairingKey(uint8_t pairingIndex,
00484     EmberKeyData *key);
00485 void emAfRf4ceGdpSetPairingKey(uint8_t pairingIndex,
00486     EmberKeyData *key);
00487 // returns true if the GDP plugin is busy doing something, false otherwise.
00488 #define emAfRf4ceGdpIsBusy() (internalGdpState() != INTERNAL_STATE_NONE)
00489 bool emAfRf4ceGdpSecurityGetRandomString(
00490     EmberAfRf4ceGdpRand *rn);
00491 void emAfRf4ceGdpSecurityValidationCompleteCallback
00492     (uint8_t pairingIndex);
00493 void emAfRf4ceGdpAttributesInitCallback(void)
00494     ;
00495 void emAfRf4ceGdpIncomingGenericResponse(
00496     EmberAfRf4ceGdpResponseCode responseCode);
00497 void emAfRf4ceGdpIncomingConfigurationComplete
00498     (EmberAfRf4ceGdpStatus status);
00499 void emAfRf4ceGdpIncomingHeartbeat(
00500     EmberAfRf4ceGdpHeartbeatTrigger trigger);
00501 void emAfRf4ceGdpIncomingGetAttributes(void);
00502 void emAfRf4ceGdpIncomingGetAttributesResponse
00503     (void);
00504 void emAfRf4ceGdpIncomingPushAttributes(void)
00505     ;
00506 void emAfRf4ceGdpIncomingSetAttributes(void);
00507 void emAfRf4ceGdpIncomingPullAttributes(void)
00508     ;
00509 void emAfRf4ceGdpIncomingPullAttributesResponse
00510     (void);
00511 void emAfRf4ceGdpIncomingCheckValidationRequest
00512     (uint8_t control);
00513 void emAfRf4ceGdpIncomingCheckValidationResponse
00514     (EmberAfRf4ceGdpCheckValidationStatus status
00515     );
00516 void emAfRf4ceGdpCheckValidationResponseSent
00517     (EmberStatus status);
00518 void emAfRf4ceGdpHeartbeatSent(EmberStatus status);
00519 void emAfRf4ceGdpIncomingKeyExchangeChallenge
00520     (EmberAfRf4ceGdpKeyExchangeFlags flags,
00521         const EmberAfRf4ceGdpRand
00522         *randA);
00523 void emAfRf4ceGdpIncomingKeyExchangeChallengeResponse
00524     (EmberAfRf4ceGdpKeyExchangeFlags flags,
00525         const EmberAfRf4ceGdpRand
00526         *randB,
00527         const EmberAfRf4ceGdpTag
00528         *tagB);
00529 void emAfRf4ceGdpIncomingKeyExchangeResponse
00530     (const EmberAfRf4ceGdpTag *tagA);
00531 void emAfRf4ceGdpKeyExchangeResponseSent(
00532     EmberStatus status);
00533 void emAfRf4ceGdpIncomingKeyExchangeConfirm
00534     (bool secured);
00535 void emAfRf4ceGdpIncomingClientNotification
00536     (EmberAfRf4ceGdpClientNotificationSubtype
00537         subType,
00538             const uint8_t *
00539             clientNotificationPayload,
00540                 uint8_t
00541                 clientNotificationPayloadLength);
00542 void emAfRf4ceZrcIncomingGenericResponse(
00543     EmberAfRf4ceGdpResponseCode responseCode);
00544 void emAfRf4ceZrcIncomingConfigurationComplete
00545     (EmberAfRf4ceGdpStatus status);
00546 void emAfRf4ceZrcIncomingGetAttributes(void);
00547 void emAfRf4ceZrcIncomingGetAttributesResponse
00548     (void);
00549 void emAfRf4ceZrcIncomingPushAttributes(void)
00550     ;
00551 void emAfRf4ceZrcIncomingSetAttributes(void);
00552 void emAfRf4ceZrcIncomingPullAttributes(void)
00553     ;

```

```

00524 void emAfRf4ceZrcIncomingPullAttributesResponse
00525     (void);
00526 void emAfRf4ceZrcIncomingClientNotification
00527     (EmberAfRf4ceGdpClientNotificationSubtype
00528         subType,
00529             const uint8_t *
00530             clientNotificationPayload,
00531                 uint8_t
00532                 clientNotificationPayloadLength);
00533
00529 bool emAfRf4ceGdpHasAttributeRecord(void);
00530 bool emAfRf4ceGdpAppendAttributeIdentificationRecord
00531     (const EmberAfRf4ceGdpAttributeIdentificationRecord
00532         *record);
00533 bool emAfRf4ceGdpFetchAttributeIdentificationRecord
00534     (EmberAfRf4ceGdpAttributeIdentificationRecord
00535         *record);
00536 bool emAfRf4ceGdpAppendAttributeStatusRecord
00537     (const EmberAfRf4ceGdpAttributeStatusRecord
00538         *record);
00539 bool emAfRf4ceGdpFetchAttributeStatusRecord
00540     (EmberAfRf4ceGdpAttributeStatusRecord *
00541         record);
00542 bool emAfRf4ceGdpAppendAttributeRecord(const
00543     EmberAfRf4ceGdpAttributeRecord *record);
00544 bool emAfRf4ceGdpFetchAttributeRecord(
00545     EmberAfRf4ceGdpAttributeRecord *record);
00546 void emAfRf4ceGdpResetFetchAttributeFinger
00547     (void);
00548 void emAfRf4ceGdpStartAttributesCommand(
00549     EmberAfRf4ceGdpCommandCode commandCode);
00550 EmberStatus emAfRf4ceGdpSendAttributesCommand(
00551     uint8_t pairingIndex,
00552             uint8_t profileId,
00553                 uint16_t vendorId);
00554
00555 EmberStatus emAfRf4ceGdpSetDiscoveryResponseAppInfo
00556     (bool pushButton,
00557             uint8_t gdpVersion);
00558 EmberStatus emAfRf4ceGdpSetPairResponseAppInfo
00559     (const EmberRf4ceApplicationInfo *pairRequestAppInfo);
00560 EmberStatus emAfRf4ceGdpSendProfileSpecificCommand
00561     (uint8_t pairingIndex,
00562             uint8_t profileId,
00563                 uint16_t vendorId,
00564                     EmberRf4ceTxOption txOptions
00565             ,
00566             uint8_t commandId,
00567                 uint8_t *commandPayload,
00568                     uint8_t commandPayloadLength
00569             ,
00570             uint8_t *messageTag);
00571
00572 EmberStatus emAfRf4ceGdpKeyExchangeChallenge(
00573     uint8_t pairingIndex,
00574             uint16_t vendorId,
00575                 EmberAfRf4ceGdpKeyExchangeFlags
00576             flags,
00577                 const EmberAfRf4ceGdpRand
00578             *randA);
00579 EmberStatus emAfRf4ceGdpKeyExchangeChallengeResponse
00580     (uint8_t pairingIndex,
00581             uint16_t vendorId,
00582                 EmberAfRf4ceGdpKeyExchangeFlags flags,
00583                     const EmberAfRf4ceGdpRand
00584             *randB,
00585                     const EmberAfRf4ceGdpTag
00586             *tagB);
00587 EmberStatus emAfRf4ceGdpKeyExchangeResponse(
00588     uint8_t pairingIndex,
00589             uint16_t vendorId,
00590                 const EmberAfRf4ceGdpTag
00591             *tagA);
00592 EmberStatus emAfRf4ceGdpKeyExchangeConfirm(
00593     uint8_t pairingIndex,

```

```

00571                               uint16_t vendorId);
00572
00573 // Host stuff
00574
00575 // Vendor ID filter (2 bytes) + minMaxClassFilter (1 byte)
00576 // + minLqiFilter (1 byte)
00577 #define GDP_SET_VALUE_BINDING_ORIGINATOR_PARAMETERS_BYTES_LENGTH
00578     4
00579 // Primary class descriptor (1 byte) + secondary class descriptor (1 byte)
00580 // + tertiary class descriptor (1 byte) + discovery LQI threshold (1 byte)
00581 #define GDP_SET_VALUE_BINDING_RECIPIENT_PARAMETERS_BYTES_LENGTH
00582     4
00583 #define GDP_SET_VALUE_FLAG_LENGTH
00584     1
00585 void emAfRf4ceGdpSetPushButtonPendingReceivedFlag
00586 (bool set);
00587 void emAfRf4ceGdpSetProxyBindingFlag(bool set);
00588 // Starts the blackout timer, turns the radio off and sets the internal state
00589 // the passed state.
00590 void emAfGdpStartBlackoutTimer(uint8_t state);
00591
00592 // Starts the command pending timer, turns the radio on and sets the internal
00593 // state the the passed state.
00594 void emAfGdpStartCommandPendingTimer(uint8_t
00595     state, uint16_t timeMs);
00596 extern uint8_t emAfRf4ceGdpOutgoingCommandFrameControl
00597 ;
00598 #define emAfRf4ceGdpOutgoingCommandsSetPendingFlag()
00599 \
00600     (emAfRf4ceGdpOutgoingCommandFrameControl |=
00601     GDP_HEADER_FRAME_CONTROL_DATA_PENDING_MASK)
00602 #define emAfRf4ceGdpOutgoingCommandsClearPendingFlag()
00603 \
00604     (emAfRf4ceGdpOutgoingCommandFrameControl &=
00605     ~GDP_HEADER_FRAME_CONTROL_DATA_PENDING_MASK)
00606
00607 #if defined(EMBER_SCRIPTED_TEST)
00608 #include "stack/core/ember-stack.h"
00609 #include "core/scripted-stub.h"
00610
00611 #define debugDiscoveryResponseDrop(reason)
00612 \
00613     simpleScriptCheck("discoveryResponseDrop", "discoveryResponseDrop: " reason,
00614     "")
00615
00616 #define debugCandidateAdded(reason)
00617 \
00618     simpleScriptCheck("candidateAdded", "candidateAdded: " reason, "")
00619
00620 #define debugScriptCheck(reason)
00621 \
00622     simpleScriptCheck("scriptCheck", "scriptCheck: " reason, "")
00623
00624 void setBindOriginatorState(uint8_t state);
00625 #else
00626 #define debugDiscoveryResponseDrop(reason)
00627 #define debugCandidateAdded(reason)
00628 #define debugScriptCheck(reason)
00629
00630 #endif // EMBER_SCRIPTED_TEST

```

8.16 rf4ce-gdp-poll.h File Reference

Macros

- #define POLLING_NEGOTIATION_PROCEDURE_DELAY_MSEC
- #define POLLING_NEGOTIATION_PROCEDURE_AFTER_FAILURE_DELAY_SEC
- #define POLLING_NEGOTIATION_PROCEDURE_CLIENT_MAX_RETRIES
- #define POLLING_NEGOTIATION_PROCEDURE_SERVER_PULL_TIMEOUT_MSEC

- #define `emAfRf4ceGdpIsPollTriggerValid(trigger)`

Functions

- void `emAfRf4ceGdpPollingStackStatusCallback` (`EmberStatus status`)
- void `emAfRf4ceGdpPollingNotifyBindingComplete` (`uint8_t pairingIndex`)
- void `emAfRf4ceGdpPollingIncomingCommandCallback` (`bool framePending`)
- void `emAfRf4ceGdpGetPollConfigurationAttribute` (`uint8_t pairingIndex, uint8_t *pollConfiguration`)
- void `emAfRf4ceGdpSetPollConfigurationAttribute` (`uint8_t pairingIndex, const uint8_t *pollConfiguration`)

8.16.1 Macro Definition Documentation

8.16.1.1 #define POLLING_NEGOTIATION_PROCEDURE_DELAY_MSEC

Definition at line 5 of file `rf4ce-gdp-poll.h`.

8.16.1.2 #define POLLING_NEGOTIATION_PROCEDURE_AFTER_FAILURE_DELAY_SEC

Definition at line 9 of file `rf4ce-gdp-poll.h`.

8.16.1.3 #define POLLING_NEGOTIATION_PROCEDURE_CLIENT_MAX_RETRIES

Definition at line 13 of file `rf4ce-gdp-poll.h`.

8.16.1.4 #define POLLING_NEGOTIATION_PROCEDURE_SERVER_PULL_TIMEOUT_MSEC

Definition at line 17 of file `rf4ce-gdp-poll.h`.

8.16.1.5 #define emAfRf4ceGdpIsPollTriggerValid(*trigger*)

Definition at line 31 of file `rf4ce-gdp-poll.h`.

8.16.2 Function Documentation

8.16.2.1 void emAfRf4ceGdpPollingStackStatusCallback (`EmberStatus status`)

8.16.2.2 void emAfRf4ceGdpPollingNotifyBindingComplete (`uint8_t pairingIndex`)

8.16.2.3 void emAfRf4ceGdpPollingIncomingCommandCallback (`bool framePending`)

8.16.2.4 void emAfRf4ceGdpGetPollConfigurationAttribute (`uint8_t pairingIndex, uint8_t *pollConfiguration`)

8.16.2.5 void emAfRf4ceGdpSetPollConfigurationAttribute (`uint8_t pairingIndex, const uint8_t *pollConfiguration`)

8.17 rf4ce-gdp-poll.h

```

00001 // Copyright 2014 Silicon Laboratories, Inc.
00002
00003 // We delay the procedure in case the enhanced security procedure takes place,
00004 // which should happen right after binding has completed.
00005 #define POLLING_NEGOTIATION_PROCEDURE_DELAY_MSEC      500
00006
00007 // If the poll negotiation procedure fails, we wait a longer delay before
00008 // trying
00009 // again.
00010 #define POLLING_NEGOTIATION_PROCEDURE_AFTER_FAILURE_DELAY_SEC     60
00011
00012 // This defines how many times the poll negotiation procedure is started again
00013 // after it failed.
00014 #define POLLING_NEGOTIATION_PROCEDURE_CLIENT_MAX_RETIRES      3
00015
00016 // The server will wait for this time for a PullAttributes() from the server
00017 // prior to declaring the poll negotiation procedure a failure.
00018 #define POLLING_NEGOTIATION_PROCEDURE_SERVER_PULL_TIMEOUT_MSEC    500
00019
00020 extern void emAfRf4ceGdpPollingStackStatusCallback
00021 (EmberStatus status);
00022
00023 extern void emAfRf4ceGdpPollingNotifyBindingComplete
00024 (uint8_t pairingIndex);
00025
00026 extern void emAfRf4ceGdpPollingIncomingCommandCallback
00027 (bool framePending,
00028 uint8_t * pairingIndex,
00029 const uint8_t * pollConfiguration);
00030
00031 #define emAfRf4ceGdpIsPollTriggerValid(trigger)
00032 \
00033     (trigger <=
EMBER_AF_RF4CE_GDP_HEARTBEAT_TRIGGER_POLLING_ON_OTHER_USER_ACTIVITY)

```

8.18 rf4ce-gdp-test.h File Reference

```
#include "rf4ce-gdp-types.h"
```

Macros

- #define EMBER_AF_PLUGIN_RF4CE_PROFILE_VENDOR_ID
- #define EMBER_AF_RF4CE_NODE_TYPE_TARGET
- #define EMBER_AF_PLUGIN_RF4CE_GDP_ENHANCED_SECURITY
- #define EMBER_AF_PLUGIN_RF4CE_GDP_STANDARD_SHARED_SECRET
- #define EMBER_AF_PLUGIN_RF4CE_GDP_VENDOR_SPECIFIC_SECRETS
- #define EMBER_AF_PLUGIN_RF4CE_GDP_POLL_SUPPORT
- #define EMBER_AF_PLUGIN_RF4CE_GDP_EXTENDED_VALIDATION
- #define EMBER_AF_PLUGIN_RF4CE_GDP_PRIMARY_CLASS_NUMBER
- #define EMBER_AF_PLUGIN_RF4CE_GDP_SECONDARY_CLASS_NUMBER
- #define EMBER_AF_PLUGIN_RF4CE_GDP_TERTIARY_CLASS_NUMBER
- #define EMBER_AF_PLUGIN_RF4CE_GDP_PRIMARY_CLASS_DUPLICATE_HANDLING

- #define EMBER_AF_PLUGIN_RF4CE_GDP_SECONDARY_CLASS_DUPLICATE_HANDLING
- #define EMBER_AF_PLUGIN_RF4CE_GDP_TERTIARY_CLASS_DUPLICATE_HANDLING
- #define EMBER_AF_PLUGIN_RF4CE_GDP_DISCOVERY_RESPONSE_LQI_THRESHOLD
- #define EMBER_AF_PLUGIN_RF4CE_GDP_VENDOR_ID_FILTER
- #define EMBER_AF_PLUGIN_RF4CE_GDP_MIN_CLASS_FILTER
- #define EMBER_AF_PLUGIN_RF4CE_GDP_MAX_CLASS_FILTER
- #define EMBER_AF_PLUGIN_RF4CE_GDP_MIN_LQI_FILTER
- #define EMBER_AF_PLUGIN_RF4CE_GDP_MIN_POLLING_INTERVAL_MS
- #define EMBER_AF_PLUGIN_RF4CE_GDP_MAX_POLLING_INTERVAL_MS
- #define EMBER_AF_PLUGIN_RF4CE_GDP_IDENTIFICATION_CAPABILITIES
- #define EMBER_AF_PLUGIN_RF4CE_GDP_APPLICATION_SPECIFIC_USER_STRING
- #define EMBER_AF_RF4CE_GDP_ATTRIBUTE_SETTABLE_SCALAR_TEST_1
- #define GDP_ATTRIBUTE_SETTABLE_SCALAR_TEST_1_SIZE
- #define GDP_ATTRIBUTE_SCALAR_TEST_1_DEFAULT
- #define SCALAR_SETTABLE_ATTRIBUTE_TEST_1
- #define EMBER_AF_RF4CE_GDP_ATTRIBUTE_SETTABLE_SCALAR_TEST_2
- #define GDP_ATTRIBUTE_SETTABLE_SCALAR_TEST_2_SIZE
- #define GDP_ATTRIBUTE_SCALAR_TEST_2_DEFAULT
- #define EMBER_AF_RF4CE_GDP_ATTRIBUTE_ONE_DIMENSIONAL_ARRAY_TEST
- #define GDP_ATTRIBUTE_ONE_DIMENSIONAL_ARRAY_TEST_DEFAULT
- #define ONE_DIMENSION_ARRAYED_ATTRIBUTE_TEST
- #define EMBER_AF_RF4CE_GDP_ATTRIBUTE_TWO_DIMENSIONAL_ARRAY_TEST
- #define GDP_ATTRIBUTE_TWO_DIMENSIONAL_ARRAY_TEST_DEFAULT
- #define GDP_ATTRIBUTE_TWO_DIMENSIONAL_ARRAY_TEST_DIMENSION
- #define TWO_DIMENSION_ARRAYED_ATTRIBUTE_TEST
- #define SCALAR_SETTABLE_ATTRIBUTE_TEST_2

Functions

- bool `emberAfPluginRf4ceGdpZrc20StartConfigurationCallback` (bool isOriginator, uint8_t pairingIndex)
- void `emberAfPluginRf4ceGdpBindingCompleteCallback` (EmberStatus status, uint8_t pairingIndex)
- void `emberAfPluginRf4ceGdpZrc20BindingCompleteCallback` (`EmberAfRf4ceGdpBindingStatus` status, uint8_t pairingIndex)
- void `emberAfPluginRf4ceGdpKeyExchangeCompleteCallback` (uint8_t pairingIndex)
- void `emberAfPluginRf4ceGdpStartValidationCallback` (uint8_t pairingIndex)
- void `emberAfPluginRf4ceGdpIdentifyCallback` (`EmberAfRf4ceGdpClientNotificationIdentifyFlags` flags, uint16_t timeS)
- bool `emberAfPluginRf4ceGdpIncomingBindProxyCallback` (const EmberEUI64 sourceIeeeAddr)
- bool `emberAfPluginRf4ceGdpVendorSpecificKeyExchangeCallback` (uint8_t initiatorVendorSpecificParam, uint8_t *responderVendorSpecificParam, uint8_t *sharedSecret)
- void `emberAfPluginRf4ceGdpHeartbeatPollingEstablishedCallback` (uint8_t pairingIndex, `EmberAfRf4ceGdpHeartbeatTrigger` pollingTriggers)
- void `emberAfPluginRf4ceGdpIdentifyClientFoundCallback` (`EmberAfRf4ceGdpClientNotificationIdentifyFlags` flags)

8.18.1 Macro Definition Documentation

8.18.1.1 `#define EMBER_AF_PLUGIN_RF4CE_PROFILE_VENDOR_ID`

Definition at line 7 of file [rf4ce-gdp-test.h](#).

8.18.1.2 `#define EMBER_AF_RF4CE_NODE_TYPE_TARGET`

Definition at line 8 of file [rf4ce-gdp-test.h](#).

8.18.1.3 `#define EMBER_AF_PLUGIN_RF4CE_GDP_ENHANCED_SECURITY`

Definition at line 9 of file [rf4ce-gdp-test.h](#).

8.18.1.4 `#define EMBER_AF_PLUGIN_RF4CE_GDP_STANDARD_SHARED_SECRET`

Definition at line 10 of file [rf4ce-gdp-test.h](#).

8.18.1.5 `#define EMBER_AF_PLUGIN_RF4CE_GDP_VENDOR_SPECIFIC_SECRETS`

Definition at line 11 of file [rf4ce-gdp-test.h](#).

8.18.1.6 `#define EMBER_AF_PLUGIN_RF4CE_GDP_POLL_SUPPORT`

Definition at line 12 of file [rf4ce-gdp-test.h](#).

8.18.1.7 `#define EMBER_AF_PLUGIN_RF4CE_GDP_EXTENDED_VALIDATION`

Definition at line 13 of file [rf4ce-gdp-test.h](#).

8.18.1.8 `#define EMBER_AF_PLUGIN_RF4CE_GDP_PRIMARY_CLASS_NUMBER`

Definition at line 15 of file [rf4ce-gdp-test.h](#).

8.18.1.9 `#define EMBER_AF_PLUGIN_RF4CE_GDP_SECONDARY_CLASS_NUMBER`

Definition at line 16 of file [rf4ce-gdp-test.h](#).

8.18.1.10 `#define EMBER_AF_PLUGIN_RF4CE_GDP_TERTIARY_CLASS_NUMBER`

Definition at line 17 of file [rf4ce-gdp-test.h](#).

8.18.1.11 `#define EMBER_AF_PLUGIN_RF4CE_GDP_PRIMARY_CLASS_DUPLICATE_HANDLING`

Definition at line 19 of file [rf4ce-gdp-test.h](#).

8.18.1.12 #define EMBER_AF_PLUGIN_RF4CE_GDP_SECONDARY_CLASS_DUPLICATE_HANDLING

Definition at line 20 of file [rf4ce-gdp-test.h](#).

8.18.1.13 #define EMBER_AF_PLUGIN_RF4CE_GDP_TERTIARY_CLASS_DUPLICATE_HANDLING

Definition at line 21 of file [rf4ce-gdp-test.h](#).

8.18.1.14 #define EMBER_AF_PLUGIN_RF4CE_GDP_DISCOVERY_RESPONSE_LQI_THRESHOLD

Definition at line 23 of file [rf4ce-gdp-test.h](#).

8.18.1.15 #define EMBER_AF_PLUGIN_RF4CE_GDP_VENDOR_ID_FILTER

Definition at line 24 of file [rf4ce-gdp-test.h](#).

8.18.1.16 #define EMBER_AF_PLUGIN_RF4CE_GDP_MIN_CLASS_FILTER

Definition at line 25 of file [rf4ce-gdp-test.h](#).

8.18.1.17 #define EMBER_AF_PLUGIN_RF4CE_GDP_MAX_CLASS_FILTER

Definition at line 26 of file [rf4ce-gdp-test.h](#).

8.18.1.18 #define EMBER_AF_PLUGIN_RF4CE_GDP_MIN_LQI_FILTER

Definition at line 27 of file [rf4ce-gdp-test.h](#).

8.18.1.19 #define EMBER_AF_PLUGIN_RF4CE_GDP_MIN_POLLING_INTERVAL_MS

Definition at line 29 of file [rf4ce-gdp-test.h](#).

8.18.1.20 #define EMBER_AF_PLUGIN_RF4CE_GDP_MAX_POLLING_INTERVAL_MS

Definition at line 30 of file [rf4ce-gdp-test.h](#).

8.18.1.21 #define EMBER_AF_PLUGIN_RF4CE_GDP_IDENTIFICATION_CAPABILITIES

Definition at line 32 of file [rf4ce-gdp-test.h](#).

8.18.1.22 #define EMBER_AF_PLUGIN_RF4CE_GDP_APPLICATION_SPECIFIC_USER_STRING

Definition at line 34 of file [rf4ce-gdp-test.h](#).

8.18.1.23 `#define EMBER_AF_RF4CE_GDP_ATTRIBUTE_SETTABLE_SCALAR_TEST_1`

Definition at line 66 of file [rf4ce-gdp-test.h](#).

8.18.1.24 `#define GDP_ATTRIBUTE_SETTABLE_SCALAR_TEST_1_SIZE`

Definition at line 67 of file [rf4ce-gdp-test.h](#).

8.18.1.25 `#define GDP_ATTRIBUTE_SCALAR_TEST_1_DEFAULT`

Definition at line 68 of file [rf4ce-gdp-test.h](#).

8.18.1.26 `#define SCALAR_SETTABLE_ATTRIBUTE_TEST_1`

Definition at line 70 of file [rf4ce-gdp-test.h](#).

8.18.1.27 `#define EMBER_AF_RF4CE_GDP_ATTRIBUTE_SETTABLE_SCALAR_TEST_2`

Definition at line 77 of file [rf4ce-gdp-test.h](#).

8.18.1.28 `#define GDP_ATTRIBUTE_SETTABLE_SCALAR_TEST_2_SIZE`

Definition at line 78 of file [rf4ce-gdp-test.h](#).

8.18.1.29 `#define GDP_ATTRIBUTE_SCALAR_TEST_2_DEFAULT`

Definition at line 79 of file [rf4ce-gdp-test.h](#).

8.18.1.30 `#define EMBER_AF_RF4CE_GDP_ATTRIBUTE_ONE_DIMENSIONAL_ARRAY_TEST`

Definition at line 81 of file [rf4ce-gdp-test.h](#).

8.18.1.31 `#define GDP_ATTRIBUTE_ONE_DIMENSIONAL_ARRAY_TEST_DEFAULT`

Definition at line 82 of file [rf4ce-gdp-test.h](#).

8.18.1.32 `#define ONE_DIMENSION_ARRAYED_ATTRIBUTE_TEST`

Definition at line 84 of file [rf4ce-gdp-test.h](#).

8.18.1.33 `#define EMBER_AF_RF4CE_GDP_ATTRIBUTE_TWO_DIMENSIONAL_ARRAY_TEST`

Definition at line 91 of file [rf4ce-gdp-test.h](#).

8.18.1.34 #define GDP_ATTRIBUTE_TWO_DIMENSIONAL_ARRAY_TEST_DEFAULT

Definition at line 92 of file [rf4ce-gdp-test.h](#).

8.18.1.35 #define GDP_ATTRIBUTE_TWO_DIMENSIONAL_ARRAY_TEST_DIMENSION

Definition at line 93 of file [rf4ce-gdp-test.h](#).

8.18.1.36 #define TWO_DIMENSION_ARRAYED_ATTRIBUTE_TEST

Definition at line 97 of file [rf4ce-gdp-test.h](#).

8.18.1.37 #define SCALAR_SETTABLE_ATTRIBUTE_TEST_2

Definition at line 105 of file [rf4ce-gdp-test.h](#).

8.18.2 Function Documentation

8.18.2.1 void emberAfPluginRf4ceGdpBindingCompleteCallback (EmberStatus status, uint8_t pairingIndex)

8.18.2.2 void emberAfPluginRf4ceGdpKeyExchangeCompleteCallback (uint8_t pairingIndex)

8.18.2.3 void emberAfPluginRf4ceGdpHeartbeatPollingEstablishedCallback (uint8_t pairingIndex, EmberAfRf4ceGdpHeartbeatTrigger pollingTriggers)

8.19 rf4ce-gdp-test.h

```

00001 // defines that are generated by app framework.
00002 // Forward declarations to avoid warnings in scripted tests (usually found in
00003 // the callback header file, that also generated by app framework).
00004
00005 #include "rf4ce-gdp-types.h"
00006
00007 #define EMBER_AF_PLUGIN_RF4CE_PROFILE_VENDOR_ID
00008     0xABCD
00009 #define EMBER_AF_PLUGIN_RF4CE_NODE_TYPE_TARGET
00010 #define EMBER_AF_PLUGIN_RF4CE_GDP_ENHANCED_SECURITY
00011 #define EMBER_AF_PLUGIN_RF4CE_GDP_STANDARD_SHARED_SECRET
00012 #define EMBER_AF_PLUGIN_RF4CE_GDP_VENDOR_SPECIFIC_SECRETS
00013 #define EMBER_AF_PLUGIN_RF4CE_GDP_POLL_SUPPORT POLL_CLIENT
00014 #define EMBER_AF_PLUGIN_RF4CE_GDP_EXTENDED_VALIDATION
00015 #define EMBER_AF_PLUGIN_RF4CE_GDP_PRIMARY_CLASS_NUMBER          0x02
00016 #define EMBER_AF_PLUGIN_RF4CE_GDP_SECONDARY_CLASS_NUMBER        0x03
00017 #define EMBER_AF_PLUGIN_RF4CE_GDP_TERTIARY_CLASS_NUMBER         0x04
00018
00019 #define EMBER_AF_PLUGIN_RF4CE_GDP_PRIMARY_CLASS_DUPLICATE_HANDLING 0x00
00020 #define EMBER_AF_PLUGIN_RF4CE_GDP_SECONDARY_CLASS_DUPLICATE_HANDLING 0x00
00021 #define EMBER_AF_PLUGIN_RF4CE_GDP_TERTIARY_CLASS_DUPLICATE_HANDLING 0x00
00022
00023 #define EMBER_AF_PLUGIN_RF4CE_GDP_DISCOVERY_RESPONSE_LQI_THRESHOLD 0x00
00024 #define EMBER_AF_PLUGIN_RF4CE_GDP_VENDOR_ID_FILTER
00025     0xFFFF
00026 #define EMBER_AF_PLUGIN_RF4CE_GDP_MIN_CLASS_FILTER             0x00
00027 #define EMBER_AF_PLUGIN_RF4CE_GDP_MAX_CLASS_FILTER             0x0E
00028 #define EMBER_AF_PLUGIN_RF4CE_GDP_MIN_LQI_FILTER              0x00
00029 #define EMBER_AF_PLUGIN_RF4CE_GDP_MIN_POLLING_INTERVAL_MS    5000
00030 #define EMBER_AF_PLUGIN_RF4CE_GDP_MAX_POLLING_INTERVAL_MS   60000
00031

```

```

00032 #define EMBER_AF_PLUGIN_RF4CE_GDP_IDENTIFICATION_CAPABILITIES          0x08
00033
00034 #define EMBER_AF_PLUGIN_RF4CE_GDP_APPLICATION_SPECIFIC_USER_STRING
00035     {0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00}
00036
00037 bool emberAfPluginRf4ceGdpZrc20StartConfigurationCallback
00038     (bool isOriginator,
00039         uint8_t
00040         pairingIndex);
00041
00042 void emberAfPluginRf4ceGdpBindingCompleteCallback
00043     (EmberStatus status,
00044         uint8_t pairingIndex);
00045
00046 void emberAfPluginRf4ceGdpKeyExchangeCompleteCallback
00047     (uint8_t pairingIndex);
00048
00049 void emberAfPluginRf4ceGdpStartValidationCallback
00050     (uint8_t pairingIndex);
00051
00052 void emberAfPluginRf4ceGdpIdentifyCallback
00053     (EmberAfRf4ceGdpClientNotificationIdentifyFlags
00054         flags,
00055         uint16_t timeS);
00056
00057 bool emberAfPluginRf4ceGdpIncomingBindProxyCallback
00058     (const EmberEUI64 sourceIeeeAddr);
00059
00060 bool emberAfPluginRf4ceGdpVendorSpecificKeyExchangeCallback
00061     (uint8_t initiatorVendorSpecificParam,
00062         uint8_t *
00063             responderVendorSpecificParam,
00064             uint8_t*
00065                 sharedSecret);
00066
00067 void emberAfPluginRf4ceGdpHeartbeatPollingEstablishedCallback
00068     (uint8_t pairingIndex,
00069         EmberAfRf4ceGdpHeartbeatTrigger pollingTriggers)
00070 ;
00071
00072 void emberAfPluginRf4ceGdpIdentifyClientFoundCallback
00073     (EmberAfRf4ceGdpClientNotificationIdentifyFlags
00074         flags);
00075
00076 // Test attributes definitions
00077
00078 #define EMBER_AF_RF4CE_GDP_ATTRIBUTE_SETTABLE_SCALAR_TEST_1          0x10
00079 #define GDP_ATTRIBUTE_SETTABLE_SCALAR_TEST_1_SIZE                      2
00080 #define GDP_ATTRIBUTE_SCALAR_TEST_1_DEFAULT                         0x5555
00081
00082 #define SCALAR_SETTABLE_ATTRIBUTE_TEST_1
00083     \
00084     {EMBER_AF_RF4CE_GDP_ATTRIBUTE_SETTABLE_SCALAR_TEST_1,
00085         \
00086         GDP_ATTRIBUTE_SETTABLE_SCALAR_TEST_1_SIZE,
00087         \
00088         (ATTRIBUTE_HAS_REMOTE_GET_ACCESS_BIT
00089             \
00090             | ATTRIBUTE_HAS_REMOTE_SET_ACCESS_BIT),
00091         \
00092         0}
00093
00094 #define EMBER_AF_RF4CE_GDP_ATTRIBUTE_SETTABLE_SCALAR_TEST_2          0x11
00095 #define GDP_ATTRIBUTE_SETTABLE_SCALAR_TEST_2_SIZE                      2
00096 #define GDP_ATTRIBUTE_SCALAR_TEST_2_DEFAULT                         0x6666
00097
00098 #define EMBER_AF_RF4CE_GDP_ATTRIBUTE_ONE_DIMENSIONAL_ARRAY_TEST      0x90
00099 #define GDP_ATTRIBUTE_ONE_DIMENSIONAL_ARRAY_TEST_DEFAULT            0xBBB
00100
00101 #define ONE_DIMENSION_ARRAYED_ATTRIBUTE_TEST
00102     \
00103     {EMBER_AF_RF4CE_GDP_ATTRIBUTE_ONE_DIMENSIONAL_ARRAY_TEST,
00104         \
00105         GDP_ATTRIBUTE_ONE_DIMENSIONAL_ARRAY_TEST_SIZE,
00106         \
00107         0}

```

```

00087 \
00088 | (ATTRIBUTE_HAS_REMOTE_GET_ACCESS_BIT
00089 | | ATTRIBUTE_HAS_REMOTE_SET_ACCESS_BIT),
00090 GDP_ATTRIBUTE_ONE_DIMENSIONAL_ARRAY_TEST_DIMENSION}
00091 #define EMBER_AF_RF4CE_GDP_ATTRIBUTE_TWO_DIMENSIONAL_ARRAY_TEST      0x91
00092 #define GDP_ATTRIBUTE_TWO_DIMENSIONAL_ARRAY_TEST_DEFAULT
00093 #define GDP_ATTRIBUTE_TWO_DIMENSIONAL_ARRAY_TEST_DIMENSION
00094 \
00095 | (GDP_ATTRIBUTE_TWO_DIMENSIONAL_ARRAY_TEST_FIRST_DIMENSION
00096 | | (GDP_ATTRIBUTE_TWO_DIMENSIONAL_ARRAY_TEST_SECOND_DIMENSION << 8))
00097 #define TWO_DIMENSION_ARRAYED_ATTRIBUTE_TEST
00098 {EMBER_AF_RF4CE_GDP_ATTRIBUTE_TWO_DIMENSIONAL_ARRAY_TEST,
00099 GDP_ATTRIBUTE_TWO_DIMENSIONAL_ARRAY_TEST_SIZE,
00100 (ATTRIBUTE_HAS_REMOTE_GET_ACCESS_BIT
00101 | ATTRIBUTE_HAS_REMOTE_SET_ACCESS_BIT
00102 | ATTRIBUTE_IS_TWO_DIMENSIONAL_ARRAYED),
00103 GDP_ATTRIBUTE_TWO_DIMENSIONAL_ARRAY_TEST_DIMENSION}
00104
00105 #define SCALAR_SETTABLE_ATTRIBUTE_TEST_2
00106 {EMBER_AF_RF4CE_GDP_ATTRIBUTE_SETTABLE_SCALAR_TEST_2,
00107 GDP_ATTRIBUTE_SETTABLE_SCALAR_TEST_2_SIZE,
00108 (ATTRIBUTE_HAS_REMOTE_GET_ACCESS_BIT
00109 | ATTRIBUTE_HAS_REMOTE_SET_ACCESS_BIT),
00110 0}
00111
00112 #if defined (POLLING_NEGOTIATION_PROCEDURE_CLIENT_MAX_RETIRES)
00113 #undef POLLING_NEGOTIATION_PROCEDURE_CLIENT_MAX_RETIRES
00114 #define POLLING_NEGOTIATION_PROCEDURE_CLIENT_MAX_RETIRES 10
00115 #endif

```

8.20 rf4ce-gdp-tokens.h File Reference

```
#include "rf4ce-gdp-attributes.h"
```

Macros

- #define **CREATOR_PLUGIN_RF4CE_GDP_BIND_TABLE**
- #define **CREATOR_PLUGIN_RF4CE_GDP_PAIRING_KEY_TABLE**
- #define **CREATOR_PLUGIN_RF4CE_GDP_POLLING_CONFIGURATION_TABLE**

8.20.1 Macro Definition Documentation

8.20.1.1 #define CREATOR_PLUGIN_RF4CE_GDP_BIND_TABLE

Definition at line 8 of file **rf4ce-gdp-tokens.h**.

8.20.1.2 #define CREATOR_PLUGIN_RF4CE_GDP_PAIRING_KEY_TABLE

Definition at line 11 of file [rf4ce-gdp-tokens.h](#).

8.20.1.3 #define CREATOR_PLUGIN_RF4CE_GDP_POLLING_CONFIGURATION_TABLE

Definition at line 13 of file [rf4ce-gdp-tokens.h](#).

8.21 rf4ce-gdp-tokens.h

```

00001 // Copyright 2014 Silicon Laboratories, Inc.
00002
00003 #include "rf4ce-gdp-attributes.h"
00004
00005 // For each pairing entry we maintain a status byte initialized to 0x00. This
00006 // is
00007 // shared between originator and recipient code (which can run both at once on
00008 // a
00009 // device).
00008 #define CREATOR_PLUGIN_RF4CE_GDP_BIND_TABLE 0x8730
00009 // For each pairing entry we need to remember the original link key that was
00010 // established during pairing.
00011 #define CREATOR_PLUGIN_RF4CE_GDP_PAIRING_KEY_TABLE 0x8731
00012 // We maintain the polling configuration attribute for each pairing entry.
00013 #define CREATOR_PLUGIN_RF4CE_GDP_POLLING_CONFIGURATION_TABLE 0x8733
00014
00015 #ifndef DEFINETYPES
00016 typedef uint8_t tokTypePairingKey[EMBER_ENCRYPTION_KEY_SIZE];
00017 typedef uint8_t tokTypePollConfiguration[APL_GDP_POLL_CONFIGURATION_SIZE
];
00018 #endif // DEFINETYPES
00019
00020 #ifndef DEFINETOKENS
00021 DEFINE_INDEXED_TOKEN(PLUGIN_RF4CE_GDP_BIND_TABLE,
00022             uint8_t,
00023             EMBER_RF4CE_PAIRING_TABLE_SIZE,
00024             0x00)
00025 DEFINE_INDEXED_TOKEN(PLUGIN_RF4CE_GDP_PAIRING_KEY_TABLE,
00026             tokTypePairingKey,
00027             EMBER_RF4CE_PAIRING_TABLE_SIZE,
00028             {0,})
00029 DEFINE_INDEXED_TOKEN(PLUGIN_RF4CE_GDP_POLLING_CONFIGURATION_TABLE,
00030             tokTypePollConfiguration,
00031             EMBER_RF4CE_PAIRING_TABLE_SIZE,
00032             {0,})
00033 #endif // DEFINETOKENS

```

8.22 rf4ce-gdp-types.h File Reference

Data Structures

- struct [EmberAfRf4ceGdpAttributeIdentificationRecord](#)
RF4CE GDP attribute identification record for Get Attributes and Pull Attributes messages.
- struct [EmberAfRf4ceGdpAttributeStatusRecord](#)
RF4CE GDP attribute identification record for Get Attributes Response and Pull Attributes Response messages.
- struct [EmberAfRf4ceGdpAttributeRecord](#)
RF4CE GDP attribute identification record for Set Attributes and Push Attributes messages.
- struct [EmberAfRf4ceGdpRand](#)
This data structure contains the GDP random byte string that is passed into various other functions.

- struct [EmberAfRf4ceGdpTag](#)

This data structure contains the GDP tag value that is passed into various other functions.

Macros

- #define [EMBER_AF_RF4CE_GDP_RAND_SIZE](#)
- #define [EMBER_AF_RF4CE_GDP_TAG_SIZE](#)

Typedefs

- typedef void(* [EmberAfRf4ceGdpHeartbeatCallback](#))(uint8_t, [EmberAfRf4ceGdpHeartbeatTrigger](#))

Enumerations

- enum [EmberAfRf4ceGdpCommandCode](#) {
 EMBER_AF_RF4CE_GDP_COMMAND_GENERIC_RESPONSE, EMBER_AF_RF4CE_GDP_COMMAND_CONFIGURATION_COMPLETE, EMBER_AF_RF4CE_GDP_COMMAND_HEARTBEAT, EMBER_AF_RF4CE_GDP_COMMAND_GET_ATTRIBUTES, EMBER_AF_RF4CE_GDP_COMMAND_GET_ATTRIBUTES_RESPONSE, EMBER_AF_RF4CE_GDP_COMMAND_PUSH_ATTRIBUTES, EMBER_AF_RF4CE_GDP_COMMAND_SET_ATTRIBUTES, EMBER_AF_RF4CE_GDP_COMMAND_PULL_ATTRIBUTES, EMBER_AF_RF4CE_GDP_COMMAND_PULL_ATTRIBUTES_RESPONSE, EMBER_AF_RF4CE_GDP_COMMAND_CHECK_VALIDATION, EMBER_AF_RF4CE_GDP_COMMAND_CLIENT_NOTIFICATION, EMBER_AF_RF4CE_GDP_COMMAND_KEY_EXCHANGE
 }
- enum [EmberAfRf4ceGdpResponseCode](#) {
 EMBER_AF_RF4CE_GDP_RESPONSE_CODE_SUCCESSFUL, EMBER_AF_RF4CE_GDP_RESPONSE_CODE_UNSUPPORTED_REQUEST, EMBER_AF_RF4CE_GDP_RESPONSE_CODE_INVALID_PARAMETER, EMBER_AF_RF4CE_GDP_RESPONSE_CODE_CONFIGURATION_FAILURE
 }
- enum [EmberAfRf4ceGdpStatus](#) {
 EMBER_AF_RF4CE_GDP_STATUS_SUCCESSFUL, EMBER_AF_RF4CE_GDP_STATUS_CONFIGURATION_FAILURE
 }
- enum [EmberAfRf4ceGdpHeartbeatTrigger](#) {
 EMBER_AF_RF4CE_GDP_HEARTBEAT_TRIGGER_GENERIC_ACTIVITY, EMBER_AF_RF4CE_GDP_HEARTBEAT_TRIGGER_TIME_BASED_POLLING, EMBER_AF_RF4CE_GDP_HEARTBEAT_TRIGGER_POLLING_ON_KEY_PRESS, EMBER_AF_RF4CE_GDP_HEARTBEAT_TRIGGER_POLLING_ON_PICKUP, EMBER_AF_RF4CE_GDP_HEARTBEAT_TRIGGER_POLLING_ON_RESET, EMBER_AF_RF4CE_GDP_HEARTBEAT_TRIGGER_POLLING_ON_MICROPHONE_ACTIVITY, EMBER_AF_RF4CE_GDP_HEARTBEAT_TRIGGER_POLLING_ON_OTHER_USER_ACTIVITY
 }
- enum [EmberAfRf4ceGdpAttributeId](#) {
 EMBER_AF_RF4CE_GDP_ATTRIBUTE_VERSION, EMBER_AF_RF4CE_GDP_ATTRIBUTE_CAPABILITIES, EMBER_AF_RF4CE_GDP_ATTRIBUTE_KEY_EXCHANGE_TRANSFER_COUNT, EMBER_AF_RF4CE_GDP_ATTRIBUTE_POWER_STATUS, EMBER_AF_RF4CE_GDP_ATTRIBUTE_POLL_CONSTRAINTS, EMBER_AF_RF4CE_GDP_ATTRIBUTE_POLL_CONFIGURATION, EMBER_AF_RF4CE_GDP_ATTRIBUTE_MAX_PAIRING_CANDIDATES, EMBER_AF_RF4CE_GDP_ATTRIBUTE_AUTO_CHECK_VALIDATION_PERIOD, EMBER_AF_RF4CE_GDP_ATTRIBUTE_BINDING_RECIPIENT_VALIDATION_WAIT_TIME, EMBER_AF_RF4CE_GDP_ATTRIBUTE_BINDING_INITIATOR_VALIDATION_WAIT_TIME, EMBER_AF_RF4CE_GDP_ATTRIBUTE_LINK_LOST_WAIT_TIME, EMBER_AF_RF4CE_GDP_ATTRIBUTE_IDENTIFICATION_CAPABILITIES
 }

- enum `EmberAfRf4ceGdpAttributeStatus` {

 `EMBER_AF_RF4CE_GDP_ATTRIBUTE_STATUS_SUCCESS`, `EMBER_AF_RF4CE_GDP_ATTRIBUTE_STATUS_UNSUPPORTED_ATTRIBUTE`, `EMBER_AF_RF4CE_GDP_ATTRIBUTE_STATUS_ILLEGAL_REQUEST`, `EMBER_AF_RF4CE_GDP_ATTRIBUTE_STATUS_INVALID_ID_ENTRY`,

 `EMBER_AF_RF4CE_GDP_ATTRIBUTE_STATUS_NO_RESPONSE`, `EMBER_AF_RF4CE_GDP_ATTRIBUTE_STATUS_INVALID_RESPONSE` }
- enum `EmberAfRf4ceGdpCheckValidationSubtype` { `EMBER_AF_RF4CE_GDP_CHECK_VALIDATION_SUBTYPE_REQUEST`, `EMBER_AF_RF4CE_GDP_CHECK_VALIDATION_SUBTYPE_RESPONSE` }
- enum `EmberAfRf4ceGdpCheckValidationStatus` { `EMBER_AF_RF4CE_GDP_CHECK_VALIDATION_STATUS_SUCCESS`, `EMBER_AF_RF4CE_GDP_CHECK_VALIDATION_STATUS_PENDING`, `EMBER_AF_RF4CE_GDP_CHECK_VALIDATION_STATUS_TIMEOUT`, `EMBER_AF_RF4CE_GDP_CHECK_VALIDATION_STATUS_FAILURE` }
- enum `EmberAfRf4ceGdpClientNotificationSubtype` { `EMBER_AF_RF4CE_GDP_CLIENT_NOTIFICATION_SUBTYPE_IDENTIFY`, `EMBER_AF_RF4CE_GDP_CLIENT_NOTIFICATION_SUBTYPE_REQUEST_POLL_NEGOTIATION` }
- enum `EmberAfRf4ceGdpClientNotificationIdentifyFlags` { `EMBER_AF_RF4CE_GDP_CLIENT_NOTIFICATION_IDENTIFY_FLAG_STOP_ON_ACTION`, `EMBER_AF_RF4CE_GDP_CLIENT_NOTIFICATION_IDENTIFY_FLAG_FLASH_LIGHT`, `EMBER_AF_RF4CE_GDP_CLIENT_NOTIFICATION_IDENTIFY_FLAG_MAKE_SOUND`, `EMBER_AF_RF4CE_GDP_CLIENT_NOTIFICATION_IDENTIFY_FLAG_VIBRATE` }
- enum `EmberAfRf4ceGdpKeyExchangeSubtype` { `EMBER_AF_RF4CE_GDP_KEY_EXCHANGE_SUBTYPE_CHALLENGE`, `EMBER_AF_RF4CE_GDP_KEY_EXCHANGE_SUBTYPE_CHALLENGE_RESPONSE`, `EMBER_AF_RF4CE_GDP_KEY_EXCHANGE_SUBTYPE_RESPONSE`, `EMBER_AF_RF4CE_GDP_KEY_EXCHANGE_SUBTYPE_CONFIRM` }
- enum `EmberAfRf4ceGdpKeyExchangeFlags` {

 `EMBER_AF_RF4CE_GDP_KEY_EXCHANGE_FLAG_STANDARD_SHARED_SECRET`, `EMBER_AF_RF4CE_GDP_KEY_EXCHANGE_FLAG_INITIATOR_VENDOR_SPECIFIC_SECRET`, `EMBER_AF_RF4CE_GDP_KEY_EXCHANGE_FLAG_RESPONDER_VENDOR_SPECIFIC_SECRET`, `EMBER_AF_RF4CE_GDP_KEY_EXCHANGE_FLAG_VENDOR_SPECIFIC_PARAMETER_MASK`,

 `EMBER_AF_RF4CE_GDP_KEY_EXCHANGE_FLAG_VENDOR_SPECIFIC_PARAMETER_OFFSET` }
- enum `EmberAfRf4ceGdpBindingState` { `EMBER_AF_RF4CE_GDP_BINDING_STATE_DORMANT`, `EMBER_AF_RF4CE_GDP_BINDING_STATE_NOT_BOUND`, `EMBER_AF_RF4CE_GDP_BINDING_STATE_BINDING`, `EMBER_AF_RF4CE_GDP_BINDING_STATE_BOUND` }
- enum `EmberAfRf4ceGdpBindingStatus` {

 `EMBER_AF_RF4CE_GDP_BINDING_STATUS_SUCCESS`, `EMBER_AF_RF4CE_GDP_BINDING_STATUS_DUPLICATE_CLASS_ABORT`, `EMBER_AF_RF4CE_GDP_BINDING_STATUS_NO_VALID_RESPONSE`, `EMBER_AF_RF4CE_GDP_BINDING_STATUS_PAIRING_FAILED`,

 `EMBER_AF_RF4CE_GDP_BINDING_STATUS_CONFIG_FAILED`, `EMBER_AF_RF4CE_GDP_BINDING_STATUS_PROFILE_SPECIFIC_CONFIG_FAILED`, `EMBER_AF_RF4CE_GDP_BINDING_STATUS_VALIDATION_FAILED` }
- enum `EmberAfRf4ceGdpPollingTrigger` {

 `EMBER_AF_RF4CE_GDP_POLLING_TRIGGER_TIME_BASED_POLLING_ENABLED`, `EMBER_AF_RF4CE_GDP_POLLING_TRIGGER_POLLING_ON_KEY_PRESS_ENABLED`, `EMBER_AF_RF4CE_GDP_POLLING_TRIGGER_POLLING_ON_PICK_UP_ENABLED`, `EMBER_AF_RF4CE_GDP_POLLING_TRIGGER_POLLING_ON_RESET_ENABLED`,

 `EMBER_AF_RF4CE_GDP_POLLING_TRIGGER_POLLING_ON_MICROPHONE_ACTIVITY_ENABLED`, `EMBER_AF_RF4CE_GDP_POLLING_TRIGGER_POLLING_ON_OTHER_USER_ACTIVITY_ENABLED` }

- enum `EmberAfRf4ceGdpPollingMethod` { `EMBER_AF_RF4CE_GDP_POLLING_METHOD_DISABLED`, `EMBER_AF_RF4CE_GDP_POLLING_METHOD_HEARTBEAT` }

8.22.1 Macro Definition Documentation

8.22.1.1 `#define EMBER_AF_RF4CE_GDP_RAND_SIZE`

Size of the GDP random byte string in bytes (8).

Definition at line 259 of file `rf4ce-gdp-types.h`.

8.22.1.2 `#define EMBER_AF_RF4CE_GDP_TAG_SIZE`

Size of the GDP tag value in in bytes (4).

Definition at line 273 of file `rf4ce-gdp-types.h`.

8.22.2 Typedef Documentation

8.22.2.1 `typedef void(* EmberAfRf4ceGdpHeartbeatCallback)(uint8_t, EmberAfRf4ceGdpHeartbeatTrigger)`

RF4CE GDP heartbeat callback. Any module can subscribe to incoming heartbeat commands by using the `emberAfRf4ceGdpSubscribeToHeartbeat()` API. The first parameter is the pairing index, the second parameter is the heartbeat trigger.

Definition at line 85 of file `rf4ce-gdp-types.h`.

8.22.3 Enumeration Type Documentation

8.22.3.1 `enum EmberAfRf4ceGdpCommandCode`

RF4CE GDP command codes.

Enumerator:

```
EMBER_AF_RF4CE_GDP_COMMAND_GENERIC_RESPONSE
EMBER_AF_RF4CE_GDP_COMMAND_CONFIGURATION_COMPLETE
EMBER_AF_RF4CE_GDP_COMMAND_HEARTBEAT
EMBER_AF_RF4CE_GDP_COMMAND_GET_ATTRIBUTES
EMBER_AF_RF4CE_GDP_COMMAND_GET_ATTRIBUTES_RESPONSE
EMBER_AF_RF4CE_GDP_COMMAND_PUSH_ATTRIBUTES
EMBER_AF_RF4CE_GDP_COMMAND_SET_ATTRIBUTES
EMBER_AF_RF4CE_GDP_COMMAND_PULL_ATTRIBUTES
EMBER_AF_RF4CE_GDP_COMMAND_PULL_ATTRIBUTES_RESPONSE
EMBER_AF_RF4CE_GDP_COMMAND_CHECK_VALIDATION
EMBER_AF_RF4CE_GDP_COMMAND_CLIENT_NOTIFICATION
EMBER_AF_RF4CE_GDP_COMMAND_KEY_EXCHANGE
```

Definition at line 10 of file `rf4ce-gdp-types.h`.

8.22.3.2 enum EmberAfRf4ceGdpResponseCode

RF4CE GDP response codes.

Enumerator:

EMBER_AF_RF4CE_GDP_RESPONSE_CODE_SUCCESSFUL
EMBER_AF_RF4CE_GDP_RESPONSE_CODE_UNSUPPORTED_REQUEST
EMBER_AF_RF4CE_GDP_RESPONSE_CODE_INVALID_PARAMETER
EMBER_AF_RF4CE_GDP_RESPONSE_CODE_CONFIGURATION_FAILURE

Definition at line 34 of file [rf4ce-gdp-types.h](#).

8.22.3.3 enum EmberAfRf4ceGdpStatus

RF4CE GDP statuses.

Enumerator:

EMBER_AF_RF4CE_GDP_STATUS_SUCCESSFUL
EMBER_AF_RF4CE_GDP_STATUS_CONFIGURATION_FAILURE

Definition at line 50 of file [rf4ce-gdp-types.h](#).

8.22.3.4 enum EmberAfRf4ceGdpHeartbeatTrigger

RF4CE GDP heartbeat triggers.

Enumerator:

EMBER_AF_RF4CE_GDP_HEARTBEAT_TRIGGER_GENERIC_ACTIVITY
EMBER_AF_RF4CE_GDP_HEARTBEAT_TRIGGER_TIME_BASED_POLLING
EMBER_AF_RF4CE_GDP_HEARTBEAT_TRIGGER_POLLING_ON_KEY_PRESS
EMBER_AF_RF4CE_GDP_HEARTBEAT_TRIGGER_POLLING_ON_PICKUP
EMBER_AF_RF4CE_GDP_HEARTBEAT_TRIGGER_POLLING_ON_RESET
EMBER_AF_RF4CE_GDP_HEARTBEAT_TRIGGER_POLLING_ON_MICROPHONE_ACTIVITY

EMBER_AF_RF4CE_GDP_HEARTBEAT_TRIGGER_POLLING_ON_OTHER_USER_ACTIVITY

Definition at line 64 of file [rf4ce-gdp-types.h](#).

8.22.3.5 enum EmberAfRf4ceGdpAttributeId

RF4CE GDP attribute ids.

Enumerator:

EMBER_AF_RF4CE_GDP_ATTRIBUTE_VERSION

EMBER_AF_RF4CE_GDP_ATTRIBUTE_CAPABILITIES
EMBER_AF_RF4CE_GDP_ATTRIBUTE_KEY_EXCHANGE_TRANSFER_COUNT
EMBER_AF_RF4CE_GDP_ATTRIBUTE_POWER_STATUS
EMBER_AF_RF4CE_GDP_ATTRIBUTE_POLL_CONSTRAINTS
EMBER_AF_RF4CE_GDP_ATTRIBUTE_POLL_CONFIGURATION
EMBER_AF_RF4CE_GDP_ATTRIBUTE_MAX_PAIRING_CANDIDATES
EMBER_AF_RF4CE_GDP_ATTRIBUTE_AUTO_CHECK_VALIDATION_PERIOD
EMBER_AF_RF4CE_GDP_ATTRIBUTE_BINDING_RECIPIENT_VALIDATION_WAIT_TIME

EMBER_AF_RF4CE_GDP_ATTRIBUTE_BINDING_INITIATOR_VALIDATION_WAIT_TIME

EMBER_AF_RF4CE_GDP_ATTRIBUTE_LINK_LOST_WAIT_TIME
EMBER_AF_RF4CE_GDP_ATTRIBUTE_IDENTIFICATION_CAPABILITIES

Definition at line 91 of file [rf4ce-gdp-types.h](#).

8.22.3.6 enum EmberAfRf4ceGdpAttributeStatus

RF4CE GDP attribute statuses.

Enumerator:

EMBER_AF_RF4CE_GDP_ATTRIBUTE_STATUS_SUCCESS
EMBER_AF_RF4CE_GDP_ATTRIBUTE_STATUS_UNSUPPORTED_ATTRIBUTE
EMBER_AF_RF4CE_GDP_ATTRIBUTE_STATUS_ILLEGAL_REQUEST
EMBER_AF_RF4CE_GDP_ATTRIBUTE_STATUS_INVALID_ENTRY
EMBER_AF_RF4CE_GDP_ATTRIBUTE_STATUS_NO_RESPONSE
EMBER_AF_RF4CE_GDP_ATTRIBUTE_STATUS_INVALID_RESPONSE

Definition at line 115 of file [rf4ce-gdp-types.h](#).

8.22.3.7 enum EmberAfRf4ceGdpCheckValidationSubtype

RF4CE GDP Check Validation subtypes.

Enumerator:

EMBER_AF_RF4CE_GDP_CHECK_VALIDATION_SUBTYPE_REQUEST
EMBER_AF_RF4CE_GDP_CHECK_VALIDATION_SUBTYPE_RESPONSE

Definition at line 167 of file [rf4ce-gdp-types.h](#).

8.22.3.8 enum EmberAfRf4ceGdpCheckValidationStatus

RF4CE GDP Check Validation statuses.

Enumerator:

*EMBER_AF_RF4CE_GDP_CHECK_VALIDATION_STATUS_SUCCESS
 EMBER_AF_RF4CE_GDP_CHECK_VALIDATION_STATUS_PENDING
 EMBER_AF_RF4CE_GDP_CHECK_VALIDATION_STATUS_TIMEOUT
 EMBER_AF_RF4CE_GDP_CHECK_VALIDATION_STATUS_FAILURE*

Definition at line 181 of file [rf4ce-gdp-types.h](#).

8.22.3.9 enum EmberAfRf4ceGdpClientNotificationSubtype

RF4CE GDP Client Notification subtypes.

Enumerator:

*EMBER_AF_RF4CE_GDP_CLIENT_NOTIFICATION_SUBTYPE_IDENTIFY
 EMBER_AF_RF4CE_GDP_CLIENT_NOTIFICATION_SUBTYPE_REQUEST_POLL_NEGOTIATION*

Definition at line 197 of file [rf4ce-gdp-types.h](#).

8.22.3.10 enum EmberAfRf4ceGdpClientNotificationIdentifyFlags

RF4CE GDP Client Notification Identify flags.

Enumerator:

*EMBER_AF_RF4CE_GDP_CLIENT_NOTIFICATION_IDENTIFY_FLAG_STOP_ON_ACTION
 EMBER_AF_RF4CE_GDP_CLIENT_NOTIFICATION_IDENTIFY_FLAG_FLASH_LIGHT
 EMBER_AF_RF4CE_GDP_CLIENT_NOTIFICATION_IDENTIFY_FLAG_MAKE_SOUND
 EMBER_AF_RF4CE_GDP_CLIENT_NOTIFICATION_IDENTIFY_FLAG_VIBRATE*

Definition at line 211 of file [rf4ce-gdp-types.h](#).

8.22.3.11 enum EmberAfRf4ceGdpKeyExchangeSubtype

RF4CE GDP Key Exchange subtypes.

Enumerator:

*EMBER_AF_RF4CE_GDP_KEY_EXCHANGE_SUBTYPE_CHALLENGE
 EMBER_AF_RF4CE_GDP_KEY_EXCHANGE_SUBTYPE_CHALLENGE_RESPONSE
 EMBER_AF_RF4CE_GDP_KEY_EXCHANGE_SUBTYPE_RESPONSE
 EMBER_AF_RF4CE_GDP_KEY_EXCHANGE_SUBTYPE_CONFIRM*

Definition at line 227 of file [rf4ce-gdp-types.h](#).

8.22.3.12 enum EmberAfRf4ceGdpKeyExchangeFlags

RF4CE GDP Key Exchange flags.

Enumerator:

```
EMBER_AF_RF4CE_GDP_KEY_EXCHANGE_FLAG_STANDARD_SHARED_SECRET
EMBER_AF_RF4CE_GDP_KEY_EXCHANGE_FLAG_INITIATOR_VENDOR_SPECIFIC_SECRET

EMBER_AF_RF4CE_GDP_KEY_EXCHANGE_FLAG_RESPONDER_VENDOR_SPECIFIC_SECRET

EMBER_AF_RF4CE_GDP_KEY_EXCHANGE_FLAG_VENDOR_SPECIFIC_PARAMETER_MASK

EMBER_AF_RF4CE_GDP_KEY_EXCHANGE_FLAG_VENDOR_SPECIFIC_PARAMETER_OFFSET
```

Definition at line 243 of file [rf4ce-gdp-types.h](#).

8.22.3.13 enum EmberAfRf4ceGdpBindingState

RF4CE GDP binding states.

Enumerator:

```
EMBER_AF_RF4CE_GDP_BINDING_STATE_DORMANT
EMBER_AF_RF4CE_GDP_BINDING_STATE_NOT_BOUND
EMBER_AF_RF4CE_GDP_BINDING_STATE_BINDING
EMBER_AF_RF4CE_GDP_BINDING_STATE_BOUND
```

Definition at line 288 of file [rf4ce-gdp-types.h](#).

8.22.3.14 enum EmberAfRf4ceGdpBindingStatus

RF4CE GDP binding statuses.

Enumerator:

```
EMBER_AF_RF4CE_GDP_BINDING_STATUS_SUCCESS
EMBER_AF_RF4CE_GDP_BINDING_STATUS_DUPLICATE_CLASS_ABORT
EMBER_AF_RF4CE_GDP_BINDING_STATUS_NO_VALID_RESPONSE
EMBER_AF_RF4CE_GDP_BINDING_STATUS_PAIRING_FAILED
EMBER_AF_RF4CE_GDP_BINDING_STATUS_CONFIG_FAILED
EMBER_AF_RF4CE_GDP_BINDING_STATUS_PROFILE_SPECIFIC_CONFIG_FAILED
EMBER_AF_RF4CE_GDP_BINDING_STATUS_VALIDATION_FAILED
```

Definition at line 304 of file [rf4ce-gdp-types.h](#).

8.22.3.15 enum EmberAfRf4ceGdpPollingTrigger

RF4CE GDP polling triggers.

Enumerator:

*EMBER_AF_RF4CE_GDP_POLLING_TRIGGER_TIME_BASED_POLLING_ENABLED
 EMBER_AF_RF4CE_GDP_POLLING_TRIGGER_POLLING_ON_KEY_PRESS_ENABLED
 EMBER_AF_RF4CE_GDP_POLLING_TRIGGER_POLLING_ON_PICK_UP_ENABLED
 EMBER_AF_RF4CE_GDP_POLLING_TRIGGER_POLLING_ON_RESET_ENABLED
 EMBER_AF_RF4CE_GDP_POLLING_TRIGGER_POLLING_ON_MICROPHONE_ACTIVITY_ENABLED
 EMBER_AF_RF4CE_GDP_POLLING_TRIGGER_POLLING_ON_OTHER_USER_ACTIVITY_ENABLED*

Definition at line 323 of file [rf4ce-gdp-types.h](#).

8.22.3.16 enum EmberAfRf4ceGdpPollingMethod

RF4CE GDP polling methods.

Enumerator:

*EMBER_AF_RF4CE_GDP_POLLING_METHOD_DISABLED
 EMBER_AF_RF4CE_GDP_POLLING_METHOD_HEARTBEAT*

Definition at line 341 of file [rf4ce-gdp-types.h](#).

8.23 rf4ce-gdp-types.h

```
00001 // Copyright 2014 Silicon Laboratories, Inc.
00002
00003 #ifndef __RF4CE_GDP_TYPES_H__
00004 #define __RF4CE_GDP_TYPES_H__
00005
00009 #ifdef DOXYGEN_SHOULD_SKIP_THIS
00010 enum EmberAfRf4ceGdpCommandCode
00011 #else
00012 typedef uint8_t EmberAfRf4ceGdpCommandCode;
00013 enum
00014 #endif
00015 {
00016     EMBER_AF_RF4CE_GDP_COMMAND_GENERIC_RESPONSE
00017         = 0x00,
00018     EMBER_AF_RF4CE_GDP_COMMAND_CONFIGURATION_COMPLETE
00019         = 0x01,
00020     EMBER_AF_RF4CE_GDP_COMMAND_HEARTBEAT
00021         = 0x02,
00022     EMBER_AF_RF4CE_GDP_COMMAND_GET_ATTRIBUTES
00023         = 0x03,
00024     EMBER_AF_RF4CE_GDP_COMMAND_GET_ATTRIBUTES_RESPONSE
00025         = 0x04,
00026     EMBER_AF_RF4CE_GDP_COMMAND_PUSH_ATTRIBUTES
00027         = 0x05,
00028     EMBER_AF_RF4CE_GDP_COMMAND_SET_ATTRIBUTES
00029         = 0x06,
00030     EMBER_AF_RF4CE_GDP_COMMAND_PULL_ATTRIBUTES
00031         = 0x07,
00032     EMBER_AF_RF4CE_GDP_COMMAND_PULL_ATTRIBUTES_RESPONSE
00033         = 0x08,
```

```

00025     EMBER_AF_RF4CE_GDP_COMMAND_CHECK_VALIDATION
00026         = 0x09,
00027     EMBER_AF_RF4CE_GDP_COMMAND_CLIENT_NOTIFICATION
00028         = 0x0A,
00029     EMBER_AF_RF4CE_GDP_COMMAND_KEY_EXCHANGE
00030         = 0x0B,
00031 };
00032 #endif DOXYGEN_SHOULD_SKIP_THIS
00033 enum EmberAfRf4ceGdpResponseCode
00034 #else
00035 typedef uint8_t EmberAfRf4ceGdpResponseCode;
00036 enum
00037 {
00038 #endif
00039 {
00040     EMBER_AF_RF4CE_GDP_RESPONSE_CODE_SUCCESSFUL
00041         = 0x00,
00042     EMBER_AF_RF4CE_GDP_RESPONSE_CODE_UNSUPPORTED_REQUEST
00043         = 0x01,
00044     EMBER_AF_RF4CE_GDP_RESPONSE_CODE_INVALID_PARAMETER
00045         = 0x02,
00046     EMBER_AF_RF4CE_GDP_RESPONSE_CODE_CONFIGURATION_FAILURE
00047         = 0x03,
00048 };
00049 #endif DOXYGEN_SHOULD_SKIP_THIS
00050 enum EmberAfRf4ceGdpStatus
00051 #else
00052 typedef uint8_t EmberAfRf4ceGdpStatus;
00053 enum
00054 #endif
00055 {
00056     EMBER_AF_RF4CE_GDP_STATUS_SUCCESSFUL
00057         = 0x00,
00058     EMBER_AF_RF4CE_GDP_STATUS_CONFIGURATION_FAILURE
00059         = 0x03,
00060 };
00061 #endif DOXYGEN_SHOULD_SKIP_THIS
00062 enum EmberAfRf4ceGdpHeartbeatTrigger
00063 #else
00064 typedef uint8_t EmberAfRf4ceGdpHeartbeatTrigger;
00065 enum
00066 #endif
00067 {
00068 #endif
00069 {
00070     EMBER_AF_RF4CE_GDP_HEARTBEAT_TRIGGER_GENERIC_ACTIVITY
00071         = 0x00,
00072     EMBER_AF_RF4CE_GDP_HEARTBEAT_TRIGGER_TIME_BASED_POLLING
00073         = 0x01,
00074     EMBER_AF_RF4CE_GDP_HEARTBEAT_TRIGGER_POLLING_ON_KEY_PRESS
00075         = 0x02,
00076     EMBER_AF_RF4CE_GDP_HEARTBEAT_TRIGGER_POLLING_ON_PICKUP
00077         = 0x03,
00078     EMBER_AF_RF4CE_GDP_HEARTBEAT_TRIGGER_POLLING_ON_RESET
00079         = 0x04,
00080     EMBER_AF_RF4CE_GDP_HEARTBEAT_TRIGGER_POLLING_ON_MICROPHONE_ACTIVITY
00081         = 0x05,
00082     EMBER_AF_RF4CE_GDP_HEARTBEAT_TRIGGER_POLLING_ON_OTHER_USER_ACTIVITY
00083         = 0x06,
00084 };
00085 typedef void (*EmberAfRf4ceGdpHeartbeatCallback
00086 ) (uint8_t, EmberAfRf4ceGdpHeartbeatTrigger);
00087 #endif DOXYGEN_SHOULD_SKIP_THIS
00088 enum EmberAfRf4ceGdpAttributeId
00089 #else
00090 typedef uint8_t EmberAfRf4ceGdpAttributeId;
00091 enum
00092 #endif
00093 {
00094 #endif
00095 {
00096     EMBER_AF_RF4CE_GDP_ATTRIBUTE_VERSION
00097         = 0x80,
00098     EMBER_AF_RF4CE_GDP_ATTRIBUTE_CAPABILITIES
00099         = 0x81,
00100     EMBER_AF_RF4CE_GDP_ATTRIBUTE_KEY_EXCHANGE_TRANSFER_COUNT
00101         = 0x82,
00102     EMBER_AF_RF4CE_GDP_ATTRIBUTE_POWER_STATUS
00103         = 0x83,
00104     EMBER_AF_RF4CE_GDP_ATTRIBUTE_POLL_CONSTRAINTS

```

```

        = 0x84,
00102 EMBER_AF_RF4CE_GDP_ATTRIBUTE_POLL_CONFIGURATION
        = 0x85,
00103 EMBER_AF_RF4CE_GDP_ATTRIBUTE_MAX_PAIRING_CANDIDATES
        = 0x86,
00104 EMBER_AF_RF4CE_GDP_ATTRIBUTE_AUTO_CHECK_VALIDATION_PERIOD
        = 0x87,
00105 EMBER_AF_RF4CE_GDP_ATTRIBUTE_BINDING_RECIPIENT_VALIDATION_WAIT_TIME
        = 0x88,
00106 EMBER_AF_RF4CE_GDP_ATTRIBUTE_BINDING_INITIATOR_VALIDATION_WAIT_TIME
        = 0x89,
00107 EMBER_AF_RF4CE_GDP_ATTRIBUTE_LINK_LOST_WAIT_TIME
        = 0x8A,
00108 EMBER_AF_RF4CE_GDP_ATTRIBUTE_IDENTIFICATION_CAPABILITIES
        = 0x8B,
00109 };
00110
00114 #ifdef DOXYGEN_SHOULD_SKIP_THIS
00115 enum EmberAfRf4ceGdpAttributeStatus
00116 #else
00117 typedef uint8_t EmberAfRf4ceGdpAttributeStatus;
00118 enum
00119 #endif
00120 {
00121     EMBER_AF_RF4CE_GDP_ATTRIBUTE_STATUS_SUCCESS
        = 0x00,
00122     EMBER_AF_RF4CE_GDP_ATTRIBUTE_STATUS_UNSUPPORTED_ATTRIBUTE
        = 0x01,
00123     EMBER_AF_RF4CE_GDP_ATTRIBUTE_STATUS_ILLEGAL_REQUEST
        = 0x02,
00124     EMBER_AF_RF4CE_GDP_ATTRIBUTE_STATUS_INVALID_ENTRY
        = 0x03,
00125     // Non over-the-air internal values.
00126     EMBER_AF_RF4CE_GDP_ATTRIBUTE_STATUS_NO_RESPONSE
        = 0xF0,
00127     EMBER_AF_RF4CE_GDP_ATTRIBUTE_STATUS_INVALID_RESPONSE
        = 0xF1,
00128 };
00129
00130
00135 typedef struct {
00136     EmberAfRf4ceGdpAttributeId attributeId;
00137     uint16_t entryId;
00138 } EmberAfRf4ceGdpAttributeIdentificationRecord
;
00139
00144 typedef struct {
00145     EmberAfRf4ceGdpAttributeId attributeId;
00146     uint16_t entryId;
00147     EmberAfRf4ceGdpAttributeStatus status;
00148     uint8_t valueLength;
00149     const uint8_t *value;
00150 } EmberAfRf4ceGdpAttributeStatusRecord;
00151
00156 typedef struct {
00157     EmberAfRf4ceGdpAttributeId attributeId;
00158     uint16_t entryId;
00159     uint8_t valueLength;
00160     const uint8_t *value;
00161 } EmberAfRf4ceGdpAttributeRecord;
00162
00166 #ifdef DOXYGEN_SHOULD_SKIP_THIS
00167 enum EmberAfRf4ceGdpCheckValidationSubtype
00168 #else
00169 typedef uint8_t EmberAfRf4ceGdpCheckValidationSubtype
;
00170 enum
00171 #endif
00172 {
00173     EMBER_AF_RF4CE_GDP_CHECK_VALIDATION_SUBTYPE_REQUEST
        = 0x00,
00174     EMBER_AF_RF4CE_GDP_CHECK_VALIDATION_SUBTYPE_RESPONSE
        = 0x01,
00175 };
00176
00180 #ifdef DOXYGEN_SHOULD_SKIP_THIS
00181 enum EmberAfRf4ceGdpCheckValidationStatus
00182 #else
00183 typedef uint8_t EmberAfRf4ceGdpCheckValidationStatus
;

```

```

00184 enum
00185 #endif
00186 {
00187     EMBER_AF_RF4CE_GDP_CHECK_VALIDATION_STATUS_SUCCESS
00188     = 0x00,
00189     EMBER_AF_RF4CE_GDP_CHECK_VALIDATION_STATUS_PENDING
00190     = 0x01,
00191     EMBER_AF_RF4CE_GDP_CHECK_VALIDATION_STATUS_TIMEOUT
00192     = 0x02,
00193     EMBER_AF_RF4CE_GDP_CHECK_VALIDATION_STATUS_FAILURE
00194     = 0x03,
00195 };
00196 #ifdef DOXYGEN_SHOULD_SKIP_THIS
00197 enum EmberAfRf4ceGdpClientNotificationSubtype
00198 #else
00199 typedef uint8_t EmberAfRf4ceGdpClientNotificationSubtype
00200 ;
00201 enum
00202 #endif
00203 {
00204     EMBER_AF_RF4CE_GDP_CLIENT_NOTIFICATION_SUBTYPE_IDENTITY
00205     = 0x00,
00206     EMBER_AF_RF4CE_GDP_CLIENT_NOTIFICATION_SUBTYPE_REQUEST_POLL_NEGOTIATION
00207     = 0x01,
00208 };
00209 #ifdef DOXYGEN_SHOULD_SKIP_THIS
00210 enum EmberAfRf4ceGdpClientNotificationIdentifyFlags
00211 #else
00212 typedef uint8_t EmberAfRf4ceGdpClientNotificationIdentifyFlags
00213 ;
00214 enum
00215 #endif
00216 {
00217     EMBER_AF_RF4CE_GDP_CLIENT_NOTIFICATION_IDENTIFY_FLAG_STOP_ON_ACTION
00218     = 0x01,
00219     EMBER_AF_RF4CE_GDP_CLIENT_NOTIFICATION_IDENTIFY_FLAG_FLASH_LIGHT
00220     = 0x02,
00221     EMBER_AF_RF4CE_GDP_CLIENT_NOTIFICATION_IDENTIFY_FLAG_MAKE_SOUND
00222     = 0x04,
00223     EMBER_AF_RF4CE_GDP_CLIENT_NOTIFICATION_IDENTIFY_FLAG_VIBRATE
00224     = 0x08,
00225 };
00226 #ifdef DOXYGEN_SHOULD_SKIP_THIS
00227 enum EmberAfRf4ceGdpKeyExchangeSubtype
00228 #else
00229 typedef uint8_t EmberAfRf4ceGdpKeyExchangeSubtype
00230 ;
00231 enum
00232 {
00233     EMBER_AF_RF4CE_GDP_KEY_EXCHANGE_SUBTYPE_CHALLENGE
00234     = 0x00,
00235     EMBER_AF_RF4CE_GDP_KEY_EXCHANGE_SUBTYPE_CHALLENGE_RESPONSE
00236     = 0x01,
00237     EMBER_AF_RF4CE_GDP_KEY_EXCHANGE_SUBTYPE_RESPONSE
00238     = 0x02,
00239     EMBER_AF_RF4CE_GDP_KEY_EXCHANGE_SUBTYPE_CONFIRM
00240     = 0x03,
00241 };
00242 #ifdef DOXYGEN_SHOULD_SKIP_THIS
00243 enum EmberAfRf4ceGdpKeyExchangeFlags
00244 #else
00245 typedef uint16_t EmberAfRf4ceGdpKeyExchangeFlags
00246 ;
00247 enum
00248 {
00249     EMBER_AF_RF4CE_GDP_KEY_EXCHANGE_FLAG_STANDARD_SHARED_SECRET
00250     = 0x0001,
00251     EMBER_AF_RF4CE_GDP_KEY_EXCHANGE_FLAG_INITIATOR_VENDOR_SPECIFIC_SECRET
00252     = 0x0002,
00253     EMBER_AF_RF4CE_GDP_KEY_EXCHANGE_FLAG_RESPONDER_VENDOR_SPECIFIC_SECRET
00254     = 0x0004,
00255     EMBER_AF_RF4CE_GDP_KEY_EXCHANGE_FLAG_VENDOR_SPECIFIC_PARAMETER_MASK
00256     = 0xFF00,
00257     EMBER_AF_RF4CE_GDP_KEY_EXCHANGE_FLAG_VENDOR_SPECIFIC_PARAMETER_OFFSET

```

```

        = 8,
00254 };
00255
00259 #define EMBER_AF_RF4CE_GDP_RAND_SIZE 8
00260
00265 typedef struct {
00267     uint8_t contents[EMBER_AF_RF4CE_GDP_RAND_SIZE];
00268 } EmberAfRf4ceGdpRand;
00269
00273 #define EMBER_AF_RF4CE_GDP_TAG_SIZE 4
00274
00279 typedef struct {
00281     uint8_t contents[EMBER_AF_RF4CE_GDP_TAG_SIZE];
00282 } EmberAfRf4ceGdpTag;
00283
00287 #ifdef DOXYGEN_SHOULD_SKIP_THIS
00288 enum EmberAfRf4ceGdpBindingState
00289#else
00290 typedef uint8_t EmberAfRf4ceGdpBindingState;
00291 enum
00292#endif
00293{
00294    EMBER_AF_RF4CE_GDP_BINDING_STATE_DORMANT
00295        = 0,
00296    EMBER_AF_RF4CE_GDP_BINDING_STATE_NOT_BOUND
00297        = 1,
00298    EMBER_AF_RF4CE_GDP_BINDING_STATE_BINDING
00299        = 2,
00300    EMBER_AF_RF4CE_GDP_BINDING_STATE_BOUND
00301        = 3,
00302};
00303
00303 #ifdef DOXYGEN_SHOULD_SKIP_THIS
00304 enum EmberAfRf4ceGdpBindingStatus
00305#else
00306 typedef uint8_t EmberAfRf4ceGdpBindingStatus;
00307 enum
00308#endif
00309{
00310    EMBER_AF_RF4CE_GDP_BINDING_STATUS_SUCCESS
00311        = 0x00,
00312    EMBER_AF_RF4CE_GDP_BINDING_STATUS_DUPLICATE_CLASS_ABORT
00313        = 0x01,
00314    EMBER_AF_RF4CE_GDP_BINDING_STATUS_NO_VALID_RESPONSE
00315        = 0x02,
00316    EMBER_AF_RF4CE_GDP_BINDING_STATUS_PAIRING_FAILED
00317        = 0x03,
00318    EMBER_AF_RF4CE_GDP_BINDING_STATUS_CONFIG_FAILED
00319        = 0x04,
00320    EMBER_AF_RF4CE_GDP_BINDING_STATUS_PROFILE_SPECIFIC_CONFIG_FAILED
00321        = 0x05,
00322    EMBER_AF_RF4CE_GDP_BINDING_STATUS_VALIDATION_FAILED
00323        = 0x06,
00324};
00325
00322 #ifdef DOXYGEN_SHOULD_SKIP_THIS
00323 enum EmberAfRf4ceGdpPollingTrigger
00324#else
00325 typedef uint16_t EmberAfRf4ceGdpPollingTrigger;
00326 enum
00327#endif
00328{
00329    EMBER_AF_RF4CE_GDP_POLLING_TRIGGER_TIME_BASED POLLING_ENABLED
00330        = 0x0001,
00331    EMBER_AF_RF4CE_GDP_POLLING_TRIGGER_POLLING_ON_KEY_PRESS_ENABLED
00332        = 0x0002,
00333    EMBER_AF_RF4CE_GDP_POLLING_TRIGGER_POLLING_ON_PICK_UP_ENABLED
00334        = 0x0004,
00335    EMBER_AF_RF4CE_GDP_POLLING_TRIGGER_POLLING_ON_RESET_ENABLED
00336        = 0x0008,
00337    EMBER_AF_RF4CE_GDP_POLLING_TRIGGER_POLLING_ON_MICROPHONE_ACTIVITY_ENABLED
00338        = 0x0010,
00339    EMBER_AF_RF4CE_GDP_POLLING_TRIGGER_POLLING_ON_OTHER_USER_ACTIVITY_ENABLED
00340        = 0x0020,
00341};
00342
00340 #ifdef DOXYGEN_SHOULD_SKIP_THIS
00341 enum EmberAfRf4ceGdpPollingMethod
00342#else
00343 typedef uint8_t EmberAfRf4ceGdpPollingMethod;
00344
```

```

00344 enum
00345 #endif
00346 {
00347     EMBER_AF_RF4CE_GDP_POLLING_METHOD_DISABLED
00348         = 0x00,
00349     EMBER_AF_RF4CE_GDP_POLLING_METHOD_HEARTBEAT
00350         = 0x01,
00351 };
00350
00351 #endif // __RF4CE_GDP_TYPES_H__

```

8.24 rf4ce-gdp.h File Reference

```
#include "rf4ce-gdp-types.h"
```

Macros

- #define EMBER_AF_PLUGIN_RF4CE_GDP_IS_ORIGINATOR
- #define POLL_CLIENT
- #define POLL_SERVER
- #define IDENTIFICATION_CLIENT
- #define IDENTIFICATION_SERVER

Functions

- uint8_t * emberAfRf4ceGdpRandContents (EmberAfRf4ceGdpRand *rand)
- uint8_t * emberAfRf4ceGdpTagContents (EmberAfRf4ceGdpTag *tag)
- EmberStatus emberAfRf4ceGdpBind (uint8_t *profileIdList, uint8_t profileIdListLength, uint8_t searchDevType)
- EmberStatus emberAfRf4ceGdpProxyBind (EmberPanId panId, EmberEUI64 ieeeAddr, uint8_t *profileIdList, uint8_t profileIdListLength)
- void emberAfRf4ceGdpConfigurationProcedureComplete (bool success)
- void emberAfRf4ceGdpPushButton (bool setPending)
- void emberAfRf4ceGdpSetValidationStatus (EmberAfRf4ceGdpCheckValidationStatus status)
- EmberStatus emberAfRf4ceGdpInitiateKeyExchange (uint8_t pairingIndex)
- EmberStatus emberAfRf4ceGdpPoll (uint8_t pairingIndex, uint16_t vendorId, EmberAfRf4ceGdpHeartbeatTrigger trigger)
- bool emberAfRf4ceGdpMessageSent (uint8_t pairingIndex, uint8_t profileId, uint16_t vendorId, const uint8_t *message, uint8_t messageLength, EmberStatus status)
- bool emberAfRf4ceGdpIncomingMessage (uint8_t pairingIndex, uint8_t profileId, uint16_t vendorId, EmberRf4ceTxOption secured, const uint8_t *message, uint8_t messageLength)
- EmberStatus emberAfRf4ceGdpGetCommandTxOptions (EmberAfRf4ceGdpCommandCode commandCode, uint8_t pairingIndex, uint16_t vendorId, EmberRf4ceTxOption *txOptions)
- EmberStatus emberAfRf4ceGdpGenericResponse (uint8_t pairingIndex, uint8_t profileId, uint16_t vendorId, EmberAfRf4ceGdpResponseCode responseCode)
- EmberStatus emberAfRf4ceGdpConfigurationComplete (uint8_t pairingIndex, uint8_t profileId, uint16_t vendorId, EmberAfRf4ceGdpStatus status)
- EmberStatus emberAfRf4ceGdpHeartbeat (uint8_t pairingIndex, uint16_t vendorId, EmberAfRf4ceGdpHeartbeatTrigger trigger)
- EmberStatus emberAfRf4ceGdpGetAttributes (uint8_t pairingIndex, uint8_t profileId, uint16_t vendorId, const EmberAfRf4ceGdpAttributeIdentificationRecord *records, uint8_t recordsLength)

- EmberStatus `emberAfRf4ceGdpGetAttributesResponse` (uint8_t pairingIndex, uint8_t profileId, uint16_t vendorId, const `EmberAfRf4ceGdpAttributeStatusRecord` *records, uint8_t recordsLength)
- EmberStatus `emberAfRf4ceGdpPushAttributes` (uint8_t pairingIndex, uint8_t profileId, uint16_t vendorId, const `EmberAfRf4ceGdpAttributeRecord` *records, uint8_t recordsLength)
- EmberStatus `emberAfRf4ceGdpSetAttributes` (uint8_t pairingIndex, uint8_t profileId, uint16_t vendorId, const `EmberAfRf4ceGdpAttributeRecord` *records, uint8_t recordsLength)
- EmberStatus `emberAfRf4ceGdpPullAttributes` (uint8_t pairingIndex, uint8_t profileId, uint16_t vendorId, const `EmberAfRf4ceGdpAttributeIdentificationRecord` *records, uint8_t recordsLength)
- EmberStatus `emberAfRf4ceGdpPullAttributesResponse` (uint8_t pairingIndex, uint8_t profileId, uint16_t vendorId, const `EmberAfRf4ceGdpAttributeStatusRecord` *records, uint8_t recordsLength)
- EmberStatus `emberAfRf4ceGdpCheckValidationRequest` (uint8_t pairingIndex, uint16_t vendorId, uint8_t control)
- EmberStatus `emberAfRf4ceGdpCheckValidationResponse` (uint8_t pairingIndex, uint16_t vendorId, `EmberAfRf4ceGdpCheckValidationStatus` status)
- EmberStatus `emberAfRf4ceGdpClientNotificationIdentify` (uint8_t pairingIndex, uint16_t vendorId, `EmberAfRf4ceGdpClientNotificationIdentifyFlags` flags, uint16_t timeS)
- EmberStatus `emberAfRf4ceGdpClientNotificationRequestPollNegotiation` (uint8_t pairingIndex, uint16_t vendorId)
- EmberStatus `emberAfRf4ceGdpClientNotification` (uint8_t pairingIndex, uint8_t profileId, uint16_t vendorId, uint8_t subtype, const uint8_t *payload, uint8_t payloadLength)
- EmberStatus `emberAfRf4ceGdpSubscribeToHeartbeat` (`EmberAfRf4ceGdpHeartbeatCallback` callback)

8.25 rf4ce-gdp.h

```

00001 // Copyright 2014 Silicon Laboratories, Inc.
00002
00003 #ifndef __RF4CE_GDP_H__
00004 #define __RF4CE_GDP_H__
00005
00006 #include "rf4ce-gdp-types.h"
00007
00035 // Controllers and targets can be originators in GDP, but only targets can be
00036 // recipients.
00040 #define EMBER_AF_PLUGIN_RF4CE_GDP_IS_ORIGINATOR
00041 #ifdef EMBER_AF_RF4CE_NODE_TYPE_TARGET
00042
00043 #define EMBER_AF_PLUGIN_RF4CE_GDP_IS_RECIPIENT
00044 #endif
00045
00046 #define POLL_CLIENT 1
00047 #define POLL_SERVER 2
00048
00049 #if EMBER_AF_PLUGIN_RF4CE_GDP_POLL_SUPPORT == POLL_CLIENT
00050
00051 #define EMBER_AF_PLUGIN_RF4CE_GDP_IS_POLL_CLIENT
00052 #elif EMBER_AF_PLUGIN_RF4CE_GDP_POLL_SUPPORT == POLL_SERVER
00053
00054 #define EMBER_AF_PLUGIN_RF4CE_GDP_IS_POLL_SERVER
00055 #endif
00056
00057 #define IDENTIFICATION_CLIENT 1
00058 #define IDENTIFICATION_SERVER 2
00059
00060 #if EMBER_AF_PLUGIN_RF4CE_GDP_IDENTIFICATION_SUPPORT == IDENTIFICATION_CLIENT
00061
00062 #define EMBER_AF_PLUGIN_RF4CE_GDP_IS_IDENTIFICATION_CLIENT
00063 #elif EMBER_AF_PLUGIN_RF4CE_GDP_IDENTIFICATION_SUPPORT == IDENTIFICATION_SERVER
00064
00065 #define EMBER_AF_PLUGIN_RF4CE_GDP_IS_IDENTIFICATION_SERVER
00066 #endif
00067
00075 #ifdef DOXYGEN_SHOULD_SKIP_THIS
00076 uint8_t *emberAfRf4ceGdpRandContents(
    EmberAfRf4ceGdpRand *rand);

```

```

00077 #else
00078 #define emberAfRf4ceGdpRandContents(rand) ((rand)->contents)
00079 #endif
00080
00087 #ifdef DOXYGEN_SHOULD_SKIP_THIS
00088 uint8_t *emberAfRf4ceGdpTagContents(
00089     EmberAfRf4ceGdpTag *tag);
00090 #define emberAfRf4ceGdpTagContents(tag) ((tag)->contents)
00091 #endif
00092
00105 EmberStatus emberAfRf4ceGdpBind(uint8_t *profileIdList,
00106                                     uint8_t profileIdListLength,
00107                                     uint8_t searchDevType);
00121 EmberStatus emberAfRf4ceGdpProxyBind(EmberPanId panId,
00122                                         EmberEUI64 ieeeAddr,
00123                                         uint8_t *profileIdList,
00124                                         uint8_t profileIdListLength);
00125
00132 void emberAfRf4ceGdpConfigurationProcedureComplete
00133     (bool success);
00139 void emberAfRf4ceGdpPushButton(bool setPending);
00140
00145 void emberAfRf4ceGdpSetValidationStatus(
00146     EmberAfRf4ceGdpCheckValidationStatus status
00147 );
00146
00153 EmberStatus emberAfRf4ceGdpInitiateKeyExchange
00154     (uint8_t pairingIndex);
00168 EmberStatus emberAfRf4ceGdpPoll(uint8_t pairingIndex,
00169                                     uint16_t vendorId,
00170                                     EmberAfRf4ceGdpHeartbeatTrigger
00171     trigger);
00184 bool emberAfRf4ceGdpMessageSent(uint8_t pairingIndex,
00185                                     uint8_t profileId,
00186                                     uint16_t vendorId,
00187                                     const uint8_t *message,
00188                                     uint8_t messageLength,
00189                                     EmberStatus status);
00190
00203 bool emberAfRf4ceGdpIncomingMessage(uint8_t
00204     pairingIndex,
00205             uint8_t profileId,
00206             uint16_t vendorId,
00207             EmberRf4ceTxOption secured,
00208             const uint8_t *message,
00209             uint8_t messageLength);
00215 EmberStatus emberAfRf4ceGdpGetCommandTxOptions
00216     (EmberAfRf4ceGdpCommandCode commandCode,
00217             uint8_t pairingIndex,
00218             uint16_t vendorId,
00219             EmberRf4ceTxOption *txOptions);
00237 EmberStatus emberAfRf4ceGdpGenericResponse(
00238     uint8_t pairingIndex,
00239             uint8_t profileId,
00240             uint16_t vendorId,
00241             EmberAfRf4ceGdpResponseCode
00242     responseCode);
00259 EmberStatus emberAfRf4ceGdpConfigurationComplete
00260     (uint8_t pairingIndex,
00261             uint8_t profileId,
00262             uint16_t vendorId,
00263             EmberAfRf4ceGdpStatus
00264     status);
00276 EmberStatus emberAfRf4ceGdpHeartbeat(uint8_t
00277     pairingIndex,
00278             uint16_t vendorId,
00279             EmberAfRf4ceGdpHeartbeatTrigger
00280     trigger);
00299 EmberStatus emberAfRf4ceGdpGetAttributes(uint8_t
00300     pairingIndex,
00301             uint8_t profileId,

```

```

00301                     uint16_t vendorId,
00302                     const
00303             EmberAfRf4ceGdpAttributeIdentificationRecord
00304             *records,
00305                     uint8_t recordsLength);
00306
00324 EmberStatus emberAfRf4ceGdpGetAttributesResponse
00325             (uint8_t pairingIndex,
00326                     uint8_t profileId,
00327                     uint16_t vendorId,
00328                     const
00329             EmberAfRf4ceGdpAttributeStatusRecord *
00330             records,
00331                     uint8_t recordsLength);
00329
00349 EmberStatus emberAfRf4ceGdpPushAttributes(uint8_t
00350             pairingIndex,
00351                     uint8_t profileId,
00352                     uint16_t vendorId,
00353                     const EmberAfRf4ceGdpAttributeRecord
00354             *records,
00355                     uint8_t recordsLength);
00354
00374 EmberStatus emberAfRf4ceGdpSetAttributes(uint8_t
00375             pairingIndex,
00376                     uint8_t profileId,
00377                     uint16_t vendorId,
00378                     const EmberAfRf4ceGdpAttributeRecord
00379             *records,
00380                     uint8_t recordsLength);
00379
00399 EmberStatus emberAfRf4ceGdpPullAttributes(uint8_t
00400             pairingIndex,
00401                     uint8_t profileId,
00402                     uint16_t vendorId,
00403                     const
00404             EmberAfRf4ceGdpAttributeIdentificationRecord
00405             *records,
00406                     uint8_t recordsLength);
00404
00424 EmberStatus emberAfRf4ceGdpPullAttributesResponse
00425             (uint8_t pairingIndex,
00426                     uint8_t profileId,
00427                     uint16_t vendorId,
00428                     const
00429             EmberAfRf4ceGdpAttributeStatusRecord *
00430             records,
00431                     uint8_t recordsLength);
00429
00444 EmberStatus emberAfRf4ceGdpCheckValidationRequest
00445             (uint8_t pairingIndex,
00446                     uint16_t vendorId,
00447                     uint8_t control);
00447
00462 EmberStatus emberAfRf4ceGdpCheckValidationResponse
00463             (uint8_t pairingIndex,
00464                     uint16_t vendorId,
00465             EmberAfRf4ceGdpCheckValidationStatus status
00466         );
00465
00483 EmberStatus emberAfRf4ceGdpClientNotificationIdentify
00484             (uint8_t pairingIndex,
00485                     uint16_t vendorId,
00486             EmberAfRf4ceGdpClientNotificationIdentifyFlags
00487             flags,
00488                     uint16_t timeS);
00487
00501 EmberStatus emberAfRf4ceGdpClientNotificationRequestPollNegotiation
00502             (uint8_t pairingIndex,
00503                     uint16_t vendorId);
00503
00522 EmberStatus emberAfRf4ceGdpClientNotification(
00523             uint8_t pairingIndex,
00524                     uint8_t profileId,
00525                     uint16_t vendorId,
00526                     uint8_t subtype,
00527                     const uint8_t *payload,
00526

```

```

00527                               uint8_t payloadLength);
00528
00540 EmberStatus emberAfRf4ceGdpSubscribeToHeartbeat
    (EmberAfRf4ceGdpHeartbeatCallback callback);
00541
00542 #endif // __RF4CE_GDP_H__
00543
00544 // @} END addtogroup

```

8.26 rf4ce-mso-attributes.h File Reference

```
#include "app/framework/plugin/rf4ce-profile/rf4ce-profile.h"
```

Data Structures

- struct [EmAfRf4ceMsoAttributeDescriptor](#)
- struct [EmAfRf4ceMsoPeripheralIdEntry](#)
- struct [EmAfRf4ceMsoRibAttributes](#)

Macros

- `#define MSO_RIB_ATTRIBUTE_PERIPHERAL_IDS_LENGTH`
- `#define MSO_RIB_ATTRIBUTE_RF_STATISTICS_LENGTH`
- `#define MSO_RIB_ATTRIBUTE_VERSIONING_LENGTH`
- `#define MSO_RIB_ATTRIBUTE_BATTERY_STATUS_LENGTH`
- `#define MSO_RIB_ATTRIBUTE_SHORT_RF_RETRY_PERIOD_LENGTH`
- `#define MSO_RIB_ATTRIBUTE_VALIDATION_CONFIGURATION_LENGTH`
- `#define MSO_RIB_ATTRIBUTE_GENERAL_PURPOSE_LENGTH`
- `#define MSO_ATTRIBUTE_HAS_REMOTE_WRITE_ACCESS_BIT`
- `#define MSO_ATTRIBUTE_IS_ARRAYED_BIT`
- `#define MSO_ATTRIBUTES_COUNT`
- `#define MSO_ATTRIBUTE_VERSIONING_ENTRIES`
- `#define MSO_ATTRIBUTE_VERSIONING_INDEX_SOFTWARE`
- `#define MSO_ATTRIBUTE_VERSIONING_INDEX_HARDWARE`
- `#define MSO_ATTRIBUTE_VERSIONING_INDEX_IR_DATABASE`

Functions

- void [emAfRf4ceMsoInitAttributes](#) (void)
- EmberAfRf4ceStatus [emAfRf4ceMsoSetAttributeRequestCallback](#) (uint8_t pairingIndex, [EmberAfRf4ceMsoAttributeId](#) attributeId, uint8_t index, uint8_t valueLen, const uint8_t *value)
- EmberAfRf4ceStatus [emAfRf4ceMsoGetAttributeRequestCallback](#) (uint8_t pairingIndex, [EmberAfRf4ceMsoAttributeId](#) attributeId, uint8_t index, uint8_t *valueLen, uint8_t *value)
- void [emAfRf4ceMsoSetAttributeResponseCallback](#) ([EmberAfRf4ceMsoAttributeId](#) attributeId, uint8_t index, [EmberAfRf4ceStatus](#) status)
- void [emAfRf4ceMsoGetAttributeResponseCallback](#) ([EmberAfRf4ceMsoAttributeId](#) attributeId, uint8_t index, [EmberAfRf4ceStatus](#) status, uint8_t valueLen, const uint8_t *value)
- bool [emAfRf4ceMsoAttributeIsSupported](#) ([EmberAfRf4ceMsoAttributeId](#) attrId)
- bool [emAfRf4ceMsoAttributeIsArrayed](#) ([EmberAfRf4ceMsoAttributeId](#) attrId)
- bool [emAfRf4ceMsoAttributeIsRemotelyWritable](#) ([EmberAfRf4ceMsoAttributeId](#) attrId)

- `uint8_t emAfRf4ceMsoGetAttributeLength (EmberAfRf4ceMsoAttributeId attrId)`
- `uint8_t emAfRf4ceMsoGetArrayedAttributeDimension (EmberAfRf4ceMsoAttributeId attrId)`
- `bool emAfRf4ceMsoArrayedAttributeIndexIsValid (uint8_t pairingIndex, EmberAfRf4ceMsoAttributeId attributeId, uint8_t index, bool isGet)`
- `void emAfRf4ceMsoWriteAttributeRecipient (uint8_t pairingIndex, EmberAfRf4ceMsoAttributeId attributeId, uint8_t index, uint8_t valueLen, const uint8_t *value)`
- `void emAfRf4ceMsoReadAttributeRecipient (uint8_t pairingIndex, EmberAfRf4ceMsoAttributeId attributeId, uint8_t index, uint8_t *valueLen, uint8_t *value)`
- `uint8_t emAfRf4ceMsoAttributePeripheralIdEntryLookup (uint8_t pairingIndex, uint8_t deviceType)`
- `uint8_t emAfRf4ceMsoAttributeGetUnusedPeripheralIdEntryIndex (uint8_t pairingIndex)`

Variables

- `EmAfRf4ceMsoRibAttributes emAfRf4ceMsoLocalRibAttributes`

8.26.1 Macro Definition Documentation

8.26.1.1 `#define MSO_RIB_ATTRIBUTE_PERIPHERAL_IDS_LENGTH`

Definition at line 8 of file `rf4ce-mso-attributes.h`.

8.26.1.2 `#define MSO_RIB_ATTRIBUTE_RF_STATISTICS_LENGTH`

Definition at line 9 of file `rf4ce-mso-attributes.h`.

8.26.1.3 `#define MSO_RIB_ATTRIBUTE_VERSIONING_LENGTH`

Definition at line 10 of file `rf4ce-mso-attributes.h`.

8.26.1.4 `#define MSO_RIB_ATTRIBUTE_BATTERY_STATUS_LENGTH`

Definition at line 11 of file `rf4ce-mso-attributes.h`.

8.26.1.5 `#define MSO_RIB_ATTRIBUTE_SHORT_RF_RETRY_PERIOD_LENGTH`

Definition at line 12 of file `rf4ce-mso-attributes.h`.

8.26.1.6 `#define MSO_RIB_ATTRIBUTE_VALIDATION_CONFIGURATION_LENGTH`

Definition at line 14 of file `rf4ce-mso-attributes.h`.

8.26.1.7 `#define MSO_RIB_ATTRIBUTE_GENERAL_PURPOSE_LENGTH`

Definition at line 15 of file `rf4ce-mso-attributes.h`.

8.26.1.8 #define MSO_ATTRIBUTE_HAS_REMOTE_WRITE_ACCESS_BIT

Definition at line 18 of file [rf4ce-mso-attributes.h](#).

8.26.1.9 #define MSO_ATTRIBUTE_IS_ARRAYED_BIT

Definition at line 19 of file [rf4ce-mso-attributes.h](#).

8.26.1.10 #define MSO_ATTRIBUTES_COUNT

Definition at line 21 of file [rf4ce-mso-attributes.h](#).

8.26.1.11 #define MSO_ATTRIBUTE_VERSIONING_ENTRIES

Definition at line 24 of file [rf4ce-mso-attributes.h](#).

8.26.1.12 #define MSO_ATTRIBUTE_VERSIONING_INDEX_SOFTWARE

Definition at line 25 of file [rf4ce-mso-attributes.h](#).

8.26.1.13 #define MSO_ATTRIBUTE_VERSIONING_INDEX_HARDWARE

Definition at line 26 of file [rf4ce-mso-attributes.h](#).

8.26.1.14 #define MSO_ATTRIBUTE_VERSIONING_INDEX_IR_DATABASE

Definition at line 27 of file [rf4ce-mso-attributes.h](#).

8.26.2 Function Documentation

8.26.2.1 void emAfRf4ceMsoInitAttributes (void)

8.26.2.2 EmberAfRf4ceStatus emAfRf4ceMsoSetAttributeRequestCallback (uint8_t pairingIndex, EmberAfRf4ceMsoAttributeId attributeld, uint8_t index, uint8_t valueLen, const uint8_t * value)

8.26.2.3 EmberAfRf4ceStatus emAfRf4ceMsoGetAttributeRequestCallback (uint8_t pairingIndex, EmberAfRf4ceMsoAttributeId attributeld, uint8_t index, uint8_t * valueLen, uint8_t * value)

8.26.2.4 void emAfRf4ceMsoSetAttributeResponseCallback (EmberAfRf4ceMsoAttributeId attributeld, uint8_t index, EmberAfRf4ceStatus status)

8.26.2.5 void emAfRf4ceMsoGetAttributeResponseCallback (EmberAfRf4ceMsoAttributeId attributeld, uint8_t index, EmberAfRf4ceStatus status, uint8_t valueLen, const uint8_t * value)

8.26.2.6 bool emAfRf4ceMsoAttributeIsSupported (EmberAfRf4ceMsoAttributeId attrId)

- 8.26.2.7 bool emAfRf4ceMsoAttributeIsArrayed (EmberAfRf4ceMsoAttributeId attrId)
- 8.26.2.8 bool emAfRf4ceMsoAttributeIsRemotelyWritable (EmberAfRf4ceMsoAttributeId attrId)
- 8.26.2.9 uint8_t emAfRf4ceMsoGetAttributeLength (EmberAfRf4ceMsoAttributeId attrId)
- 8.26.2.10 uint8_t emAfRf4ceMsoGetArrayedAttributeDimension (EmberAfRf4ceMsoAttributeId attrId)
- 8.26.2.11 bool emAfRf4ceMsoArrayedAttributeIndexIsValid (uint8_t pairingIndex, EmberAfRf4ceMsoAttributeId attributeId, uint8_t index, bool isGet)
- 8.26.2.12 void emAfRf4ceMsoWriteAttributeRecipient (uint8_t pairingIndex, EmberAfRf4ceMsoAttributeId attributeId, uint8_t index, uint8_t valueLen, const uint8_t * value)
- 8.26.2.13 void emAfRf4ceMsoReadAttributeRecipient (uint8_t pairingIndex, EmberAfRf4ceMsoAttributeId attributeId, uint8_t index, uint8_t * valueLen, uint8_t * value)
- 8.26.2.14 uint8_t emAfRf4ceMsoAttributePeripheralIdEntryLookup (uint8_t pairingIndex, uint8_t deviceType)
- 8.26.2.15 uint8_t emAfRf4ceMsoAttributeGetUnusedPeripheralIdEntryIndex (uint8_t pairingIndex)

8.26.3 Variable Documentation

- 8.26.3.1 EmAfRf4ceMsoRibAttributes emAfRf4ceMsoLocalRibAttributes

8.27 rf4ce-mso-attributes.h

```

00001 // Copyright 2014 Silicon Laboratories, Inc.
00002
00003 #ifndef _RF4CE_MSO_ATTRIBUTES_H_
00004 #define _RF4CE_MSO_ATTRIBUTES_H_
00005
00006 #include "app/framework/plugin/rf4ce-profile/rf4ce-profile.h"
00007
00008 #define MSO_RIB_ATTRIBUTE_PERIPHERAL_IDS_LENGTH          4
00009 #define MSO_RIB_ATTRIBUTE_RF_STATISTICS_LENGTH           16
00010 #define MSO_RIB_ATTRIBUTE_VERSIONING_LENGTH              2
00011 #define MSO_RIB_ATTRIBUTE_BATTERY_STATUS_LENGTH         11
00012 #define MSO_RIB_ATTRIBUTE_SHORT_RF_RETRY_PERIOD_LENGTH 4
00013 // IR-RF database length has variable length
00014 #define MSO_RIB_ATTRIBUTE_VALIDATION_CONFIGURATION_LENGTH 4
00015 #define MSO_RIB_ATTRIBUTE_GENERAL_PURPOSE_LENGTH        16
00016
00017 // Bitmask field definitions.
00018 #define MSO_ATTRIBUTE_HAS_REMOTE_WRITE_ACCESS_BIT        0x01
00019 #define MSO_ATTRIBUTE_IS_ARRAYED_BIT                      0x02
00020
00021 #define MSO_ATTRIBUTES_COUNT                            8
00022
00023 // Versioning attribute
00024 #define MSO_ATTRIBUTE_VERSIONING_ENTRIES               3
00025 #define MSO_ATTRIBUTE_VERSIONING_INDEX_SOFTWARE        0x00
00026 #define MSO_ATTRIBUTE_VERSIONING_INDEX_HARDWARE        0x01
00027 #define MSO_ATTRIBUTE_VERSIONING_INDEX_IR_DATABASE      0x02
00028
00029 typedef struct {
00030     EmberAfRf4ceMsoAttributeId id;
00031     uint8_t size;
00032     uint8_t bitmask;

```

```

00033     uint8_t dimension; // only for arrayed attributes
00034 } EmAfRf4ceMsoAttributeDescriptor;
00035
00036 typedef struct {
00037     uint8_t deviceType;
00038     uint8_t peripheralId[MSO_RIB_ATTRIBUTE_PERIPHERAL_IDS_LENGTH
00039 ];
00039 } EmAfRf4ceMsoPeripheralIdEntry;
00040
00041 typedef struct {
00042     EmAfRf4ceMsoPeripheralIdEntry peripheralIds[
00043         EMBER_AF_PLUGIN_RF4CE_MSO_PERIPHERAL_ID_ENTRIES
00044     ];
00043     uint8_t rfStatistics[MSO_RIB_ATTRIBUTE_RF_STATISTICS_LENGTH
00044 ];
00044     uint8_t versioning[MSO_ATTRIBUTE_VERSIONING_ENTRIES
00045 [MSO_RIB_ATTRIBUTE_VERSIONING_LENGTH];
00045     uint8_t batteryStatus[MSO_RIB_ATTRIBUTE_BATTERY_STATUS_LENGTH
00046 ];
00046     uint8_t shortRfRetryPeriod[MSO_RIB_ATTRIBUTE_SHORT_RF_RETRY_PERIOD_LENGTH
00047 ];
00047     // IR-RF database stored by the application
00048     uint8_t validationConfiguration[
00049         MSO_RIB_ATTRIBUTE_VALIDATION_CONFIGURATION_LENGTH
00049 ];
00049     uint8_t generalPurpose[EMBER_AF_PLUGIN_RF4CE_MSO_GENERAL_PURPOSE_ENTRIES
00050 [MSO_RIB_ATTRIBUTE_GENERAL_PURPOSE_LENGTH
00050 ];
00050 } EmAfRf4ceMsoRibAttributes;
00051
00052 // Originator local attributes.
00053 extern EmAfRf4ceMsoRibAttributes
00053     emAfRf4ceMsoLocalRibAttributes;
00054
00055 void emAfRf4ceMsoInitAttributes(void);
00056
00057 EmberAfRf4ceStatus emAfRf4ceMsoSetAttributeRequestCallback
00058     (uint8_t pairingIndex,
00058     EmberAfRf4ceMsoAttributeId attributeId,
00059                                         uint8_t index,
00060                                         uint8_t valueLen,
00061                                         const uint8_t *value
00061 );
00062
00063 EmberAfRf4ceStatus emAfRf4ceMsoGetAttributeRequestCallback
00064     (uint8_t pairingIndex,
00064     EmberAfRf4ceMsoAttributeId attributeId,
00065                                         uint8_t index,
00066                                         uint8_t *valueLen,
00067                                         uint8_t *value);
00068
00069 void emAfRf4ceMsoSetAttributeResponseCallback
00070     (EmberAfRf4ceMsoAttributeId attributeId,
00070                                         uint8_t index,
00071                                         EmberAfRf4ceStatus
00071     status);
00072
00073 void emAfRf4ceMsoGetAttributeResponseCallback
00074     (EmberAfRf4ceMsoAttributeId attributeId,
00074                                         uint8_t index,
00075                                         EmberAfRf4ceStatus
00075     status,
00076                                         uint8_t valueLen,
00077                                         const uint8_t *value);
00078
00079 bool emAfRf4ceMsoAttributeIsSupported(
00080     EmberAfRf4ceMsoAttributeId attrId);
00080 bool emAfRf4ceMsoAttributeIsArrayed(
00081     EmberAfRf4ceMsoAttributeId attrId);
00081 bool emAfRf4ceMsoAttributeIsRemotelyWritable
00082     (EmberAfRf4ceMsoAttributeId attrId);
00082 uint8_t emAfRf4ceMsoGetAttributeLength(
00083     EmberAfRf4ceMsoAttributeId attrId);
00083 uint8_t emAfRf4ceMsoGetArrayedAttributeDimension
00084     (EmberAfRf4ceMsoAttributeId attrId);
00084 bool emAfRf4ceMsoArrayedAttributeIndexIsValid
00085     (uint8_t pairingIndex,
00085                                         EmberAfRf4ceMsoAttributeId

```

```

        attributeId,
00086                               uint8_t index,
00087                               bool isGet);
00088
00089 void emAfRf4ceMsoWriteAttributeRecipient(
00090     uint8_t pairingIndex,
00091                               EmberAfRf4ceMsoAttributeId
00092     attributeId,
00093                               uint8_t index,
00094                               uint8_t valueLen,
00095                               const uint8_t *value);
00096
00097 void emAfRf4ceMsoReadAttributeRecipient(
00098     uint8_t pairingIndex,
00099                               EmberAfRf4ceMsoAttributeId
00100     attributeId,
00101                               uint8_t index,
00102                               uint8_t *valueLen,
00103                               uint8_t *value);
00104
00105 uint8_t emAfRf4ceMsoAttributePeripheralIdEntryLookup
00106     (uint8_t pairingIndex,
00107      uint8_t deviceType);
00108 uint8_t emAfRf4ceMsoAttributeGetUnusedPeripheralIdEntryIndex
00109     (uint8_t pairingIndex);
00110
00111 #endif // _RF4CE_MSO_ATTRIBUTES_H_

```

8.28 rf4ce-mso-internal.h File Reference

```
#include "app/framework/plugin/rf4ce-profile/rf4ce-profile.h"
```

Data Structures

- struct EmAfMsoPairingCandidate

Macros

- #define NWK_DISCOVERY_LQI_THRESHOLD
- #define NWK_DISCOVERY_REPETITION_INTERVAL_MS
- #define NWK_INDICATE_DISCOVERY_REQUEST
- #define NWK_MAX_DISCOVERY_REPEATITIONS
- #define NWK_MAX_REPORTED_NODE_DESCRIPTOROS
- #define APLC_MAX_PAIRING_CANDIDATES
- #define APLC_MAX_KEY_REPEAT_INTERVAL_MS
- #define APLC_MAX_RIB_ATTRIBUTE_SIZE
- #define APLC_RESPONSE_IDLE_TIME_MS
- #define APLC_BLACK_OUT_TIME_MS
- #define APLC_MIN_KEY_EXCHANGE_TRANSFER_COUNT
- #define NULL_PAIRING_INDEX
- #define MSO_USER_STRING_LENGTH
- #define MSO_DISCOVERY_REQUEST_MSO_USER_STRING_OFFSET
- #define MSO_DISCOVERY_REQUEST_MSO_USER_STRING_LENGTH
- #define MSO_DISCOVERY_REQUEST_BINDING_INITIATION_INDICATOR_OFFSET
- #define MSO_BINDING_INITIATION_INDICATOR_DEDICATED_KEY_COMBO_BIND
- #define MSO_BINDING_INITIATION_INDICATOR_ANY_BUTTON_BIND
- #define MSO_DISCOVERY_RESPONSE_MSO_USER_STRING_OFFSET

- #define **MSO_DISCOVERY_RESPONSE_MS0_USER_STRING_LENGTH**
- #define **MSO_DISCOVERY_RESPONSE_TERTIARY_CLASS_DESCRIPTOR_OFFSET**
- #define **MSO_DISCOVERY_RESPONSE_SECONDARY_CLASS_DESCRIPTOR_OFFSET**
- #define **MSO_DISCOVERY_RESPONSE_PRIMARY_CLASS_DESCRIPTOR_OFFSET**
- #define **MSO_DISCOVERY_RESPONSE_STRICT_LQI_THRESHOLD_OFFSET**
- #define **MSO_DISCOVERY_RESPONSE_BASIC_LQI_THRESHOLD_OFFSET**
- #define **MSO_CLASS_DESCRIPTOR_CLASS_NUMBER_MASK**
- #define **MSO_CLASS_DESCRIPTOR_CLASS_NUMBER_OFFSET**
- #define **MSO_CLASS_DESCRIPTOR_DUPLICATE_CLASS_NUMBER_HANDLING_MASK**
- #define **MSO_CLASS_DESCRIPTOR_DUPLICATE_CLASS_NUMBER_HANDLING_OFFSET**
- #define **MSO_CLASS_DESCRIPTOR_APPLY_STRICT_LQI_THRESHOLD_BIT**
- #define **MSO_CLASS_DESCRIPTOR_ENABLE_REQUEST_AUTO_VALIDATION_BIT**
- #define **MSO_DUPLICATE_CLASS_NUMBER_HANDLING_USE_NODE_AS_IS**
- #define **MSO_DUPLICATE_CLASS_NUMBER_HANDLING_REMOVE_NODE_DESCRIPTOR**
- #define **MSO_DUPLICATE_CLASS_NUMBER_HANDLING_RECLASSIFY_NODE_DESCRIPTOR**
- #define **MSO_DUPLICATE_CLASS_NUMBER_HANDLING_ABORT_BINDING**
- #define **USE_NODE_DESCRIPTOR_AS_IS**
- #define **REMOVE_NODE_DESCRIPTOR**
- #define **RECLASSIFY_NODE_DESCRIPTOR**
- #define **ABORT_BINDING**
- #define **MSO_FULL_ROLL_BACK_ENABLED_BIT**
- #define **MSO_PRIMARY_APPLY_STRICT_LQI_THRESHOLD_BIT**
- #define **MSO_PRIMARY_ENABLE_REQUEST_AUTO_VALIDATION_BIT**
- #define **MSO_SECONDARY_APPLY_STRICT_LQI_THRESHOLD_BIT**
- #define **MSO_SECONDARY_ENABLE_REQUEST_AUTO_VALIDATION_BIT**
- #define **MSO_TERTIARY_APPLY_STRICT_LQI_THRESHOLD_BIT**
- #define **MSO_TERTIARY_ENABLE_REQUEST_AUTO_VALIDATION_BIT**
- #define **MSO_NODE_DESCRIPTOR_CONTROL_CURRENT_CLASS_DESCRIPTOR_MASK**
- #define **MSO_NODE_DESCRIPTOR_CONTROL_CURRENT_CLASS_DESCRIPTOR_OFFSET**
- #define **MSO_NODE_DESCRIPTOR_CONTROL_CURRENT_CLASS_DESCRIPTOR_PRIMARY**
- #define **MSO_NODE_DESCRIPTOR_CONTROL_CURRENT_CLASS_DESCRIPTOR_SECONDARY**
- #define **MSO_NODE_DESCRIPTOR_CONTROL_CURRENT_CLASS_DESCRIPTOR_TERTIARY**
- #define **MSO_NODE_DESCRIPTOR_CONTROL_CURRENT_CLASS_DESCRIPTOR_RESERVED**
- #define **MSO_NODE_DESCRIPTOR_CONTROL_ENTRY_IN_USE_BIT**
- #define **MSO_HEADER_LENGTH**
- #define **MSO_HEADER_FRAME_CONTROL_OFFSET**
- #define **MSO_HEADER_FRAME_CONTROL_COMMAND_CODE_MASK**
- #define **USER_CONTROL_PRESSED_LENGTH**
- #define **USER_CONTROL_PRESSED_RC_COMMAND_CODE_OFFSET**
- #define **USER_CONTROL_PRESSED_RC_COMMAND_PAYLOAD_OFFSET**
- #define **USER_CONTROL_REPEAT_1_0_LENGTH**
- #define **USER_CONTROL_REPEAT_1_1_LENGTH**
- #define **USER_CONTROL_REPEAT_1_1_RC_COMMAND_CODE_OFFSET**
- #define **USER_CONTROL_REPEAT_1_1_RC_COMMAND_PAYLOAD_OFFSET**

- #define **USER_CONTROL_RELEASED_1_0_LENGTH**
- #define **USER_CONTROL_RELEASED_1_1_LENGTH**
- #define **USER_CONTROL_RELEASED_1_1_RC_COMMAND_CODE_OFFSET**
- #define **CHECK_VALIDATION_REQUEST_LENGTH**
- #define **CHECK_VALIDATION_REQUEST_CHECK_VALIDATION_CONTROL_OFFSET**
- #define **CHECK_VALIDATION_CONTROL_REQUEST_AUTOMATIC_VALIDATION_BIT**
- #define **CHECK_VALIDATION_RESPONSE_LENGTH**
- #define **CHECK_VALIDATION_RESPONSE_CHECK_VALIDATION_STATUS_OFFSET**
- #define **SET_ATTRIBUTE_REQUEST_LENGTH**
- #define **SET_ATTRIBUTE_REQUEST_ATTRIBUTE_ID_OFFSET**
- #define **SET_ATTRIBUTE_REQUEST_INDEX_OFFSET**
- #define **SET_ATTRIBUTE_REQUEST_VALUE_LENGTH_OFFSET**
- #define **SET_ATTRIBUTE_REQUEST_VALUE_OFFSET**
- #define **SET_ATTRIBUTE_RESPONSE_LENGTH**
- #define **SET_ATTRIBUTE_RESPONSE_ATTRIBUTE_ID_OFFSET**
- #define **SET_ATTRIBUTE_RESPONSE_INDEX_OFFSET**
- #define **SET_ATTRIBUTE_RESPONSE_STATUS_OFFSET**
- #define **GET_ATTRIBUTE_REQUEST_LENGTH**
- #define **GET_ATTRIBUTE_REQUEST_ATTRIBUTE_ID_OFFSET**
- #define **GET_ATTRIBUTE_REQUEST_INDEX_OFFSET**
- #define **GET_ATTRIBUTE_REQUEST_VALUE_LENGTH_OFFSET**
- #define **GET_ATTRIBUTE_RESPONSE_LENGTH**
- #define **GET_ATTRIBUTE_RESPONSE_ATTRIBUTE_ID_OFFSET**
- #define **GET_ATTRIBUTE_RESPONSE_INDEX_OFFSET**
- #define **GET_ATTRIBUTE_RESPONSE_STATUS_OFFSET**
- #define **GET_ATTRIBUTE_RESPONSE_VALUE_LENGTH_OFFSET**
- #define **GET_ATTRIBUTE_RESPONSE_VALUE_OFFSET**
- #define **EMBER_AF_PLUGIN_RF4CE_MSO_MAXIMUM_RC_COMMAND_PAYLOAD_LENGTH**
- #define **MAXIMUM_USER_CONTROL_X_LENGTH**
- #define **EMBER_AF_PLUGIN_RF4CE_MSO_MAXIMUM_PAYLOAD_LENGTH**
- #define **MSO_SET_VALUE_BINDING_RECIPIENT_PARAMETERS_BYTES_LENGTH**
- #define **debugScriptCheck(reason)**

Enumerations

- enum {
 BINDING_STATE_GET_VALIDATION_CONFIG_PENDING, **BINDING_STATE_CHECK_VALIDATION_INITIAL**, **BINDING_STATE_CHECK_VALIDATION_SENDING_REQUEST**, **BINDING_STATE_CHECK_VALIDATION_IDLING**, **BINDING_STATE_CHECK_VALIDATION_WAIT_FOR_RESPONSE** }

Functions

- bool **emAfRf4ceMsoIsBlackedOut** (uint8_t pairingIndex)
- EmberStatus **emAfRf4ceMsoSend** (uint8_t pairingIndex)
- EmberStatus **emAfRf4ceMsoSendExtended** (uint8_t pairingIndex, EmberRf4ceTxOption txOptions)
- void **emAfRf4ceMsoMessageSent** (uint8_t pairingIndex, **EmberAfRf4ceMsoCommandCode** commandCode, const uint8_t *message, uint8_t messageLength, EmberStatus status)

- void `emAfRf4ceMsoIncomingMessage` (uint8_t pairingIndex, `EmberAfRf4ceMsoCommandCode` commandCode, const uint8_t *message, uint8_t messageLength)
- void `emAfRf4ceMsoSetValidation` (uint8_t pairingIndex, `EmberAfRf4ceMsoValidationState` state, `EmberAfRf4ceMsoCheckValidationStatus` status)
- void `emAfRf4ceMsoInitializeValidationStates` (void)
- `EmberAfRf4ceMsoValidationState emAfRf4ceMsoGetValidationState` (uint8_t pairingIndex)
- `EmberAfRf4ceMsoCheckValidationStatus emAfRf4ceMsoGetValidationStatus` (uint8_t pairingIndex)
- void `emAfPluginRf4ceMsoCheckValidationRequestSentCallback` (EmberStatus status, uint8_t pairingIndex)
- void `emAfPluginRf4ceMsoIncomingCheckValidationResponseCallback` (`EmberAfRf4ceMsoCheckValidationStatus` status, uint8_t pairingIndex)
- void `emAfRf4ceMsoValidationConfigurationResponseCallback` (`EmberAfRf4ceStatus` status)
- uint8_t `emAfRf4ceMsoGetActiveBindPairingIndex` (void)
- void `emAfRf4ceMsoSetActiveBindPairingIndex` (uint8_t pairingIndex)
- EmberStatus `emAfRf4ceMsoSetDiscoveryResponseUserString` (void)
- void `emAfRf4ceMsoInitCommands` (void)

Variables

- uint8_t `emAfRf4ceMsoBuffer` []
- uint8_t `emAfRf4ceMsoBufferLength`
- EmberEventControl `emberAfPluginRf4ceMsoUserControlEventControl`
- EmberEventControl `emberAfPluginRf4ceMsoCheckValidationEventControl`
- EmberEventControl `emberAfPluginRf4ceMsoSetGetAttributeEventControl`

8.28.1 Macro Definition Documentation

8.28.1.1 #define NWK_DISCOVERY_LQI_THRESHOLD

Definition at line 5 of file `rf4ce-mso-internal.h`.

8.28.1.2 #define NWK_DISCOVERY_REPEATITION_INTERVAL_MS

Definition at line 6 of file `rf4ce-mso-internal.h`.

8.28.1.3 #define NWK_INDICATE_DISCOVERY_REQUEST

Definition at line 7 of file `rf4ce-mso-internal.h`.

8.28.1.4 #define NWK_MAX_DISCOVERY_REPEATITIONS

Definition at line 10 of file `rf4ce-mso-internal.h`.

8.28.1.5 #define NWK_MAX_REPORTED_NODE_DESCRIPTOROS

Definition at line 13 of file `rf4ce-mso-internal.h`.

8.28.1.6 #define APLC_MAX_PAIRING_CANDIDATES

Definition at line 17 of file [rf4ce-mso-internal.h](#).

8.28.1.7 #define APLC_MAX_KEY_REPEAT_INTERVAL_MS

Definition at line 21 of file [rf4ce-mso-internal.h](#).

8.28.1.8 #define APLC_MAX_RIB_ATTRIBUTE_SIZE

Definition at line 26 of file [rf4ce-mso-internal.h](#).

8.28.1.9 #define APLC_RESPONSE_IDLE_TIME_MS

Definition at line 31 of file [rf4ce-mso-internal.h](#).

8.28.1.10 #define APLC_BLACK_OUT_TIME_MS

Definition at line 35 of file [rf4ce-mso-internal.h](#).

8.28.1.11 #define APLC_MIN_KEY_EXCHANGE_TRANSFER_COUNT

Definition at line 39 of file [rf4ce-mso-internal.h](#).

8.28.1.12 #define NULL_PAIRING_INDEX

Definition at line 41 of file [rf4ce-mso-internal.h](#).

8.28.1.13 #define MSO_USER_STRING_LENGTH

Definition at line 43 of file [rf4ce-mso-internal.h](#).

8.28.1.14 #define MSO_DISCOVERY_REQUEST_MSO_USER_STRING_OFFSET

Definition at line 46 of file [rf4ce-mso-internal.h](#).

8.28.1.15 #define MSO_DISCOVERY_REQUEST_MSO_USER_STRING_LENGTH

Definition at line 47 of file [rf4ce-mso-internal.h](#).

8.28.1.16 #define MSO_DISCOVERY_REQUEST_BINDING_INITIATION_INDICATOR_OFFSET

Definition at line 48 of file [rf4ce-mso-internal.h](#).

8.28.1.17 #define MSO_BINDING_INITIATION_INDICATOR_DEDICATED_KEY_COMBO_BIND

Definition at line 51 of file [rf4ce-mso-internal.h](#).

8.28.1.18 #define MSO_BINDING_INITIATION_INDICATOR_ANY_BUTTON_BIND

Definition at line 52 of file [rf4ce-mso-internal.h](#).

8.28.1.19 #define MSO_DISCOVERY_RESPONSE_MSO_USER_STRING_OFFSET

Definition at line 55 of file [rf4ce-mso-internal.h](#).

8.28.1.20 #define MSO_DISCOVERY_RESPONSE_MSO_USER_STRING_LENGTH

Definition at line 56 of file [rf4ce-mso-internal.h](#).

8.28.1.21 #define MSO_DISCOVERY_RESPONSE_TERTIARY_CLASS_DESCRIPTOR_OFFSET

Definition at line 57 of file [rf4ce-mso-internal.h](#).

8.28.1.22 #define MSO_DISCOVERY_RESPONSE_SECONDARY_CLASS_DESCRIPTOR_OFFSET

Definition at line 58 of file [rf4ce-mso-internal.h](#).

8.28.1.23 #define MSO_DISCOVERY_RESPONSE_PRIMARY_CLASS_DESCRIPTOR_OFFSET

Definition at line 59 of file [rf4ce-mso-internal.h](#).

8.28.1.24 #define MSO_DISCOVERY_RESPONSE_STRICT_LQI_THRESHOLD_OFFSET

Definition at line 60 of file [rf4ce-mso-internal.h](#).

8.28.1.25 #define MSO_DISCOVERY_RESPONSE_BASIC_LQI_THRESHOLD_OFFSET

Definition at line 61 of file [rf4ce-mso-internal.h](#).

8.28.1.26 #define MSO_CLASS_DESCRIPTOR_CLASS_NUMBER_MASK

Definition at line 64 of file [rf4ce-mso-internal.h](#).

8.28.1.27 #define MSO_CLASS_DESCRIPTOR_CLASS_NUMBER_OFFSET

Definition at line 65 of file [rf4ce-mso-internal.h](#).

8.28.1.28 #define MSO_CLASS_DESCRIPTOR_DUPLICATE_CLASS_NUMBER_HANDLING_MASK

Definition at line 66 of file [rf4ce-mso-internal.h](#).

8.28.1.29 #define MSO_CLASS_DESCRIPTOR_DUPLICATE_CLASS_NUMBER_HANDLING_OFFSET

Definition at line 67 of file [rf4ce-mso-internal.h](#).

8.28.1.30 #define MSO_CLASS_DESCRIPTOR_APPLY_STRICT_LQI_THRESHOLD_BIT

Definition at line 68 of file [rf4ce-mso-internal.h](#).

8.28.1.31 #define MSO_CLASS_DESCRIPTOR_ENABLE_REQUEST_AUTO_VALIDATION_BIT

Definition at line 69 of file [rf4ce-mso-internal.h](#).

8.28.1.32 #define MSO_DUPLICATE_CLASS_NUMBER_HANDLING_USE_NODE_AS_IS

Definition at line 72 of file [rf4ce-mso-internal.h](#).

8.28.1.33 #define MSO_DUPLICATE_CLASS_NUMBER_HANDLING_REMOVE_NODE_DESCRIPTOR

Definition at line 73 of file [rf4ce-mso-internal.h](#).

8.28.1.34 #define MSO_DUPLICATE_CLASS_NUMBER_HANDLING_RECLASSIFY_NODE_DESCRIPTOR

Definition at line 74 of file [rf4ce-mso-internal.h](#).

8.28.1.35 #define MSO_DUPLICATE_CLASS_NUMBER_HANDLING_ABORT_BINDING

Definition at line 75 of file [rf4ce-mso-internal.h](#).

8.28.1.36 #define USE_NODE_DESCRIPTOR_AS_IS

Definition at line 79 of file [rf4ce-mso-internal.h](#).

8.28.1.37 #define REMOVE_NODE_DESCRIPTOR

Definition at line 81 of file [rf4ce-mso-internal.h](#).

8.28.1.38 #define RECLASSIFY_NODE_DESCRIPTOR

Definition at line 83 of file [rf4ce-mso-internal.h](#).

8.28.1.39 #define ABORT_BINDING

Definition at line 85 of file [rf4ce-mso-internal.h](#).

8.28.1.40 #define MSO_FULL_ROLL_BACK_ENABLED_BIT

Definition at line 91 of file [rf4ce-mso-internal.h](#).

8.28.1.41 #define MSO_PRIMARY_APPLY_STRICT_LQI_THRESHOLD_BIT

Definition at line 98 of file [rf4ce-mso-internal.h](#).

8.28.1.42 #define MSO_PRIMARY_ENABLE_REQUEST_AUTO_VALIDATION_BIT

Definition at line 104 of file [rf4ce-mso-internal.h](#).

8.28.1.43 #define MSO_SECONDARY_APPLY_STRICT_LQI_THRESHOLD_BIT

Definition at line 111 of file [rf4ce-mso-internal.h](#).

8.28.1.44 #define MSO_SECONDARY_ENABLE_REQUEST_AUTO_VALIDATION_BIT

Definition at line 117 of file [rf4ce-mso-internal.h](#).

8.28.1.45 #define MSO_TERTIARY_APPLY_STRICT_LQI_THRESHOLD_BIT

Definition at line 124 of file [rf4ce-mso-internal.h](#).

8.28.1.46 #define MSO_TERTIARY_ENABLE_REQUEST_AUTO_VALIDATION_BIT

Definition at line 131 of file [rf4ce-mso-internal.h](#).

8.28.1.47 #define MSO_NODE_DESCRIPTOR_CONTROL_CURRENT_CLASS_DESCRIPTOR_MASK

Definition at line 135 of file [rf4ce-mso-internal.h](#).

8.28.1.48 #define MSO_NODE_DESCRIPTOR_CONTROL_CURRENT_CLASS_DESCRIPTOR_OFFSET

Definition at line 136 of file [rf4ce-mso-internal.h](#).

8.28.1.49 #define MSO_NODE_DESCRIPTOR_CONTROL_CURRENT_CLASS_DESCRIPTOR_PRIMARY

Definition at line 137 of file [rf4ce-mso-internal.h](#).

8.28.1.50 #define MSO_NODE_DESCRIPTOR_CONTROL_CURRENT_CLASS_DESCRIPTOR_SECONDARY

Definition at line 138 of file [rf4ce-mso-internal.h](#).

8.28.1.51 #define MSO_NODE_DESCRIPTOR_CONTROL_CURRENT_CLASS_DESCRIPTOR_TERTIARY

Definition at line 139 of file [rf4ce-mso-internal.h](#).

8.28.1.52 #define MSO_NODE_DESCRIPTOR_CONTROL_CURRENT_CLASS_DESCRIPTOR_RESERVED

Definition at line 140 of file [rf4ce-mso-internal.h](#).

8.28.1.53 #define MSO_NODE_DESCRIPTOR_CONTROL_ENTRY_IN_USE_BIT

Definition at line 141 of file [rf4ce-mso-internal.h](#).

8.28.1.54 #define MSO_HEADER_LENGTH

Definition at line 145 of file [rf4ce-mso-internal.h](#).

8.28.1.55 #define MSO_HEADER_FRAME_CONTROL_OFFSET

Definition at line 146 of file [rf4ce-mso-internal.h](#).

8.28.1.56 #define MSO_HEADER_FRAME_CONTROL_COMMAND_CODE_MASK

Definition at line 147 of file [rf4ce-mso-internal.h](#).

8.28.1.57 #define USER_CONTROL_PRESSED_LENGTH

Definition at line 152 of file [rf4ce-mso-internal.h](#).

8.28.1.58 #define USER_CONTROL_PRESSED_RC_COMMAND_CODE_OFFSET

Definition at line 153 of file [rf4ce-mso-internal.h](#).

8.28.1.59 #define USER_CONTROL_PRESSED_RC_COMMAND_PAYLOAD_OFFSET

Definition at line 154 of file [rf4ce-mso-internal.h](#).

8.28.1.60 #define USER_CONTROL_REPEAT_1_0_LENGTH

Definition at line 159 of file [rf4ce-mso-internal.h](#).

8.28.1.61 #define USER_CONTROL_REPEAT_1_1_LENGTH

Definition at line 160 of file [rf4ce-mso-internal.h](#).

8.28.1.62 #define USER_CONTROL_REPEAT_1_1_RC_COMMAND_CODE_OFFSET

Definition at line 161 of file [rf4ce-mso-internal.h](#).

8.28.1.63 #define USER_CONTROL_REPEAT_1_1_RC_COMMAND_PAYLOAD_OFFSET

Definition at line 162 of file [rf4ce-mso-internal.h](#).

8.28.1.64 #define USER_CONTROL_RELEASED_1_0_LENGTH

Definition at line 166 of file [rf4ce-mso-internal.h](#).

8.28.1.65 #define USER_CONTROL_RELEASED_1_1_LENGTH

Definition at line 167 of file [rf4ce-mso-internal.h](#).

8.28.1.66 #define USER_CONTROL_RELEASED_1_1_RC_COMMAND_CODE_OFFSET

Definition at line 168 of file [rf4ce-mso-internal.h](#).

8.28.1.67 #define CHECK_VALIDATION_REQUEST_LENGTH

Definition at line 172 of file [rf4ce-mso-internal.h](#).

8.28.1.68 #define CHECK_VALIDATION_REQUEST_CHECK_VALIDATION_CONTROL_OFFSET

Definition at line 173 of file [rf4ce-mso-internal.h](#).

8.28.1.69 #define CHECK_VALIDATION_CONTROL_REQUEST_AUTOMATIC_VALIDATION_BIT

Definition at line 175 of file [rf4ce-mso-internal.h](#).

8.28.1.70 #define CHECK_VALIDATION_RESPONSE_LENGTH

Definition at line 179 of file [rf4ce-mso-internal.h](#).

8.28.1.71 #define CHECK_VALIDATION_RESPONSE_CHECK_VALIDATION_STATUS_OFFSET

Definition at line 180 of file [rf4ce-mso-internal.h](#).

8.28.1.72 #define SET_ATTRIBUTE_REQUEST_LENGTH

Definition at line 187 of file [rf4ce-mso-internal.h](#).

8.28.1.73 #define SET_ATTRIBUTE_REQUEST_ATTRIBUTE_ID_OFFSET

Definition at line 188 of file [rf4ce-mso-internal.h](#).

8.28.1.74 #define SET_ATTRIBUTE_REQUEST_INDEX_OFFSET

Definition at line 189 of file [rf4ce-mso-internal.h](#).

8.28.1.75 #define SET_ATTRIBUTE_REQUEST_VALUE_LENGTH_OFFSET

Definition at line 190 of file [rf4ce-mso-internal.h](#).

8.28.1.76 #define SET_ATTRIBUTE_REQUEST_VALUE_OFFSET

Definition at line 191 of file [rf4ce-mso-internal.h](#).

8.28.1.77 #define SET_ATTRIBUTE_RESPONSE_LENGTH

Definition at line 197 of file [rf4ce-mso-internal.h](#).

8.28.1.78 #define SET_ATTRIBUTE_RESPONSE_ATTRIBUTE_ID_OFFSET

Definition at line 198 of file [rf4ce-mso-internal.h](#).

8.28.1.79 #define SET_ATTRIBUTE_RESPONSE_INDEX_OFFSET

Definition at line 199 of file [rf4ce-mso-internal.h](#).

8.28.1.80 #define SET_ATTRIBUTE_RESPONSE_STATUS_OFFSET

Definition at line 200 of file [rf4ce-mso-internal.h](#).

8.28.1.81 #define GET_ATTRIBUTE_REQUEST_LENGTH

Definition at line 206 of file [rf4ce-mso-internal.h](#).

8.28.1.82 #define GET_ATTRIBUTE_REQUEST_ATTRIBUTE_ID_OFFSET

Definition at line 207 of file [rf4ce-mso-internal.h](#).

8.28.1.83 #define GET_ATTRIBUTE_REQUEST_INDEX_OFFSET

Definition at line 208 of file [rf4ce-mso-internal.h](#).

8.28.1.84 #define GET_ATTRIBUTE_REQUEST_VALUE_LENGTH_OFFSET

Definition at line 209 of file [rf4ce-mso-internal.h](#).

8.28.1.85 #define GET_ATTRIBUTE_RESPONSE_LENGTH

Definition at line 217 of file [rf4ce-mso-internal.h](#).

8.28.1.86 #define GET_ATTRIBUTE_RESPONSE_ATTRIBUTE_ID_OFFSET

Definition at line 218 of file [rf4ce-mso-internal.h](#).

8.28.1.87 #define GET_ATTRIBUTE_RESPONSE_INDEX_OFFSET

Definition at line 219 of file [rf4ce-mso-internal.h](#).

8.28.1.88 #define GET_ATTRIBUTE_RESPONSE_STATUS_OFFSET

Definition at line 220 of file [rf4ce-mso-internal.h](#).

8.28.1.89 #define GET_ATTRIBUTE_RESPONSE_VALUE_LENGTH_OFFSET

Definition at line 221 of file [rf4ce-mso-internal.h](#).

8.28.1.90 #define GET_ATTRIBUTE_RESPONSE_VALUE_OFFSET

Definition at line 222 of file [rf4ce-mso-internal.h](#).

8.28.1.91 #define EMBER_AF_PLUGIN_RF4CE_MSO_MAXIMUM_RC_COMMAND_PAYLOAD_LENGTH

Definition at line 229 of file [rf4ce-mso-internal.h](#).

8.28.1.92 #define MAXIMUM_USER_CONTROL_X_LENGTH

Definition at line 231 of file [rf4ce-mso-internal.h](#).

8.28.1.93 #define EMBER_AF_PLUGIN_RF4CE_MSO_MAXIMUM_PAYLOAD_LENGTH

Definition at line 239 of file [rf4ce-mso-internal.h](#).

8.28.1.94 `#define MSO_SET_VALUE_BINDING_RECIPIENT_PARAMETERS_BYTES_LENGTH`

Definition at line 245 of file [rf4ce-mso-internal.h](#).

8.28.1.95 `#define debugScriptCheck(reason)`

Definition at line 314 of file [rf4ce-mso-internal.h](#).

8.28.2 Enumeration Type Documentation

8.28.2.1 anonymous enum

Enumerator:

BINDING_STATE_GET_VALIDATION_CONFIG_PENDING
BINDING_STATE_CHECK_VALIDATION_INITIAL
BINDING_STATE_CHECK_VALIDATION_SENDING_REQUEST
BINDING_STATE_CHECK_VALIDATION_IDLING
BINDING_STATE_CHECK_VALIDATION_WAIT_FOR_RESPONSE

Definition at line 248 of file [rf4ce-mso-internal.h](#).

8.28.3 Function Documentation

8.28.3.1 `bool emAfRf4ceMsolsBlackedOut (uint8_t pairingIndex)`

8.28.3.2 `EmberStatus emAfRf4ceMsoSend (uint8_t pairingIndex)`

8.28.3.3 `EmberStatus emAfRf4ceMsoSendExtended (uint8_t pairingIndex, EmberRf4ceTxOption txOptions)`

8.28.3.4 `void emAfRf4ceMsoMessageSent (uint8_t pairingIndex, EmberAfRf4ceMsoCommandCode commandCode, const uint8_t * message, uint8_t messageLength, EmberStatus status)`

8.28.3.5 `void emAfRf4ceMsolIncomingMessage (uint8_t pairingIndex, EmberAfRf4ceMsoCommandCode commandCode, const uint8_t * message, uint8_t messageLength)`

8.28.3.6 `void emAfRf4ceMsoSetValidation (uint8_t pairingIndex, EmberAfRf4ceMsoValidationState state, EmberAfRf4ceMsoCheckValidationStatus status)`

8.28.3.7 `void emAfRf4ceMsolInitializeValidationStates (void)`

8.28.3.8 `EmberAfRf4ceMsoValidationState emAfRf4ceMsoGetValidationState (uint8_t pairingIndex)`

8.28.3.9 `EmberAfRf4ceMsoCheckValidationStatus emAfRf4ceMsoGetValidationStatus (uint8_t pairingIndex)`

8.28.3.10 `void emAfPluginRf4ceMsoCheckValidationRequestSentCallback (EmberStatus status, uint8_t pairingIndex)`

- 8.28.3.11 void emAfPluginRf4ceMsoIncomingCheckValidationResponseCallback (EmberAfRf4ceMsoCheckValidationStatus *status*, uint8_t *pairingIndex*)
- 8.28.3.12 void emAfRf4ceMsoValidationConfigurationResponseCallback (EmberAfRf4ceStatus *status*)
- 8.28.3.13 uint8_t emAfRf4ceMsoGetActiveBindPairingIndex (void)
- 8.28.3.14 void emAfRf4ceMsoSetActiveBindPairingIndex (uint8_t *pairingIndex*)
- 8.28.3.15 EmberStatus emAfRf4ceMsoSetDiscoveryResponseUserString (void)
- 8.28.3.16 void emAfRf4ceMsoInitCommands (void)

8.28.4 Variable Documentation

- 8.28.4.1 uint8_t emAfRf4ceMsoBuffer[]
- 8.28.4.2 uint8_t emAfRf4ceMsoBufferLength
- 8.28.4.3 EmberEventControl emAfPluginRf4ceMsoUserControlEventControl
- 8.28.4.4 EmberEventControl emAfPluginRf4ceMsoCheckValidationEventControl
- 8.28.4.5 EmberEventControl emAfPluginRf4ceMsoSetGetAttributeEventControl

8.29 rf4ce-mso-internal.h

```

00001 // Copyright 2013 Silicon Laboratories, Inc.
00002
00003 #include "app/framework/plugin/rf4ce-profile/rf4ce-profile.h"
00004 "
00005 #define NWK_DISCOVERY_LQI_THRESHOLD 0
00006 #define NWK_DISCOVERY_REPEAT_INTERVAL_MS 600
00007 #define NWK_INDICATE_DISCOVERY_REQUEST true
00008
00009 // The maximum number of repetitions performed during discovery.
00010 #define NWK_MAX_DISCOVERY_REPETITIONS 2
00011
00012 // The maximum number of node descriptors reported during discovery.
00013 #define NWK_MAX_REPORTED_NODE_DESCRIPTOR 16
00014
00015 // The maximum number of pairing candidates selected from the node descriptor
00016 // list.
00017 #define APLC_MAX_PAIRING_CANDIDATES 3
00018
00019 // The maximum time between consecutive user control repeated command frame
00020 // transmissions.
00021 #define APLC_MAX_KEY_REPEAT_INTERVAL_MS 120
00022
00023 // The maximum size in octets of the elements of the attributes in the RIB. At
00024 // the same time, the maximum size in octets of the value field in the set
00025 // attribute request and get attribute response command frames.
00026 #define APLC_MAX_RIB_ATTRIBUTE_SIZE 92
00027
00028 // The time a device SHALL wait after the successful transmission of a request
00029 // command frame before enabling its receiver to receive a response command
00030 // frame.
00031 #define APLC_RESPONSE_IDLE_TIME_MS 50
00032
00033 // The time at the start of the validation procedure during which packets shall
00034 // not be transmitted.
00035 #define APLC_BLACK_OUT_TIME_MS 100
00036
00037 // The minimum value of the KeyExTransferCount parameter passed to the pair

```

```

00038 // request primitive during the push button pairing procedure.
00039 #define APLC_MIN_KEY_EXCHANGE_TRANSFER_COUNT 3
00040
00041 #define NULL_PAIRING_INDEX 0xFF
00042
00043 #define MSO_USER_STRING_LENGTH
00044     9
00045
00046 // Discovery request user string fields
00047 #define MSO_DISCOVERY_REQUEST_MS0_USER_STRING_OFFSET
00048     MSO_USER_STRING_LENGTH
00049 #define MSO_DISCOVERY_REQUEST_BINDING_INITIATION_INDICATOR_OFFSET
00050     14
00051
00052 // Binding initiation indicator field
00053 #define MSO_BINDING_INITIATION_INDICATOR_DEDICATED_KEY_COMBO_BIND
00054     0x00
00055 #define MSO_BINDING_INITIATION_INDICATOR_ANY_BUTTON_BIND
00056     0x01
00057
00058 // Discovery response user string fields
00059 #define MSO_DISCOVERY_RESPONSE_MS0_USER_STRING_OFFSET
00060     MSO_USER_STRING_LENGTH
00061 #define MSO_DISCOVERY_RESPONSE_TERTIARY_CLASS_DESCRIPTOR_OFFSET
00062     10
00063 #define MSO_DISCOVERY_RESPONSE_SECONDARY_CLASS_DESCRIPTOR_OFFSET
00064     11
00065 #define MSO_DISCOVERY_RESPONSE_PRIMARY_CLASS_DESCRIPTOR_OFFSET
00066     12
00067 #define MSO_DISCOVERY_RESPONSE_STRICT_LQI_THRESHOLD_OFFSET
00068     13
00069 #define MSO_DISCOVERY_RESPONSE_BASIC_LQI_THRESHOLD_OFFSET
00070     14
00071
00072 // Class descriptor fields
00073 #define MSO_CLASS_DESCRIPTOR_CLASS_NUMBER_MASK
00074     0x0F
00075 #define MSO_CLASS_DESCRIPTOR_CLASS_NUMBER_OFFSET
00076     0x30
00077 #define MSO_CLASS_DESCRIPTOR_DUPLICATE_CLASS_NUMBER_HANDLING_MASK
00078     0x40
00079 #define MSO_CLASS_DESCRIPTOR_APPLY_STRICT_LQI_THRESHOLD_BIT
00080     0x80
00081
00082 // These come from the plugin option and need to be translated to appropriate
00083 // values.
00084 #define USE_NODE_DESCRIPTOR_AS_IS \
00085     MSO_DUPLICATE_CLASS_NUMBER_HANDLING_USE_NODE_AS_IS
00086 #define REMOVE_NODE_DESCRIPTOR \
00087     MSO_DUPLICATE_CLASS_NUMBER_HANDLING_REMOVE_NODE_DESCRIPTOR
00088 #define RECLASSIFY_NODE_DESCRIPTOR \
00089     MSO_DUPLICATE_CLASS_NUMBER_HANDLING_RECLASSIFY_NODE_DESCRIPTOR
00090 #define ABORT_BINDING \
00091     MSO_DUPLICATE_CLASS_NUMBER_HANDLING_ABORT_BINDING
00092
00093 #ifdef EMBER_AF_PLUGIN_RF4CE_MSO_FULL_ROLL_BACK_ENABLED
00094     #define MSO_FULL_ROLL_BACK_ENABLED_BIT BIT(0)
00095 #else
00096     #define MSO_FULL_ROLL_BACK_ENABLED_BIT 0
00097 #endif
00098
00099 #ifdef EMBER_AF_PLUGIN_RF4CE_MSO_PRIMARY_APPLY_STRICT_LQI_THRESHOLD
00100     #define MSO_PRIMARY_APPLY_STRICT_LQI_THRESHOLD_BIT \
00101         MSO_CLASS_DESCRIPTOR_APPLY_STRICT_LQI_THRESHOLD_BIT
00102 #else
00103     #define MSO_PRIMARY_APPLY_STRICT_LQI_THRESHOLD_BIT 0

```

```

00099 #endif
00100 #ifdef EMBER_AF_PLUGIN_RF4CE_MSO_PRIMARY_ENABLE_REQUEST_AUTO_VALIDATION
00101     #define MSO_PRIMARY_ENABLE_REQUEST_AUTO_VALIDATION_BIT \
00102         MSO_CLASS_DESCRIPTOR_ENABLE_REQUEST_AUTO_VALIDATION_BIT
00103 #else
00104     #define MSO_PRIMARY_ENABLE_REQUEST_AUTO_VALIDATION_BIT 0
00105 #endif
00106
00107 #ifdef EMBER_AF_PLUGIN_RF4CE_MSO_SECONDARY_APPLY_STRICT_LQI_THRESHOLD
00108     #define MSO_SECONDARY_APPLY_STRICT_LQI_THRESHOLD_BIT \
00109         MSO_CLASS_DESCRIPTOR_APPLY_STRICT_LQI_THRESHOLD_BIT
00110 #else
00111     #define MSO_SECONDARY_APPLY_STRICT_LQI_THRESHOLD_BIT 0
00112 #endif
00113 #ifdef EMBER_AF_PLUGIN_RF4CE_MSO_SECONDARY_ENABLE_REQUEST_AUTO_VALIDATION
00114     #define MSO_SECONDARY_ENABLE_REQUEST_AUTO_VALIDATION_BIT \
00115         MSO_CLASS_DESCRIPTOR_ENABLE_REQUEST_AUTO_VALIDATION_BIT
00116 #else
00117     #define MSO_SECONDARY_ENABLE_REQUEST_AUTO_VALIDATION_BIT 0
00118 #endif
00119
00120 #ifdef EMBER_AF_PLUGIN_RF4CE_MSO_TERTIARY_APPLY_STRICT_LQI_THRESHOLD
00121     #define MSO_TERTIARY_APPLY_STRICT_LQI_THRESHOLD_BIT \
00122         MSO_CLASS_DESCRIPTOR_APPLY_STRICT_LQI_THRESHOLD_BIT
00123 #else
00124     #define MSO_TERTIARY_APPLY_STRICT_LQI_THRESHOLD_BIT 0
00125 #endif
00126
00127 #ifdef EMBER_AF_PLUGIN_RF4CE_MSO_TERTIARY_ENABLE_REQUEST_AUTO_VALIDATION
00128     #define MSO_TERTIARY_ENABLE_REQUEST_AUTO_VALIDATION_BIT \
00129         MSO_CLASS_DESCRIPTOR_ENABLE_REQUEST_AUTO_VALIDATION_BIT
00130 #else
00131     #define MSO_TERTIARY_ENABLE_REQUEST_AUTO_VALIDATION_BIT 0
00132 #endif
00133
00134 // Internal node descriptor control byte.
00135 #define MSO_NODE_DESCRIPTOR_CONTROL_CURRENT_CLASS_DESCRIPTOR_MASK
0x03
00136 #define MSO_NODE_DESCRIPTOR_CONTROL_CURRENT_CLASS_DESCRIPTOR_OFFSET
0
00137 #define MSO_NODE_DESCRIPTOR_CONTROL_CURRENT_CLASS_DESCRIPTOR_PRIMARY
0x00
00138 #define MSO_NODE_DESCRIPTOR_CONTROL_CURRENT_CLASS_DESCRIPTOR_SECONDARY
0x01
00139 #define MSO_NODE_DESCRIPTOR_CONTROL_CURRENT_CLASS_DESCRIPTOR_TERTIARY
0x02
00140 #define MSO_NODE_DESCRIPTOR_CONTROL_CURRENT_CLASS_DESCRIPTOR_RESERVED
0x03
00141 #define MSO_NODE_DESCRIPTOR_CONTROL_ENTRY_IN_USE_BIT
0x04
00142
00143 // MSO header
00144 // - Frame control (1 byte)
00145 #define MSO_HEADER_LENGTH 1
00146 #define MSO_HEADER_FRAME_CONTROL_OFFSET 0
00147 #define MSO_HEADER_FRAME_CONTROL_COMMAND_CODE_MASK 0xFF
00148
00149 // User Control Pressed
00150 // - RC command code (1 byte)
00151 // - RC command payload (n bytes)
00152 #define USER_CONTROL_PRESSED_LENGTH (MSO_HEADER_LENGTH + 1)
00153 #define USER_CONTROL_PRESSED_RC_COMMAND_CODE_OFFSET (MSO_HEADER_LENGTH)
00154 #define USER_CONTROL_PRESSED_RC_COMMAND_PAYLOAD_OFFSET (MSO_HEADER_LENGTH + 1)
00155
00156 // User Control Repeated
00157 // - RC command code (1 byte, 1.1 only)
00158 // - RC command payload (n bytes, 1.1 only)
00159 #define USER_CONTROL_REPEATED_1_0_LENGTH (MSO_HEADER_LENGTH)
00160 #define USER_CONTROL_REPEATED_1_1_LENGTH (MSO_HEADER_LENGTH + 1)
00161 #define USER_CONTROL_REPEATED_1_1_RC_COMMAND_CODE_OFFSET (MSO_HEADER_LENGTH)
00162 #define USER_CONTROL_REPEATED_1_1_RC_COMMAND_PAYLOAD_OFFSET (MSO_HEADER_LENGTH
+ 1)
00163
00164 // User Control Released
00165 // - RC command code (1 byte, 1.1 only)
00166 #define USER_CONTROL_RELEASED_1_0_LENGTH (MSO_HEADER_LENGTH)
00167 #define USER_CONTROL_RELEASED_1_1_LENGTH (MSO_HEADER_LENGTH + 1)
00168 #define USER_CONTROL_RELEASED_1_1_RC_COMMAND_CODE_OFFSET (MSO_HEADER_LENGTH)
00169
00170 // Check Validation Request

```

```

00171 // - Check validation control (1 byte)
00172 #define CHECK_VALIDATION_REQUEST_LENGTH (MSO_HEADER_LENGTH + 1)
00173 #define CHECK_VALIDATION_REQUEST_CHECK_VALIDATION_CONTROL_OFFSET
    (MSO_HEADER_LENGTH)
00174
00175 #define CHECK_VALIDATION_CONTROL_REQUEST_AUTOMATIC_VALIDATION_BIT
    0x01
00176
00177 // Check Validation Response
00178 // - Check validation status (1 byte)
00179 #define CHECK_VALIDATION_RESPONSE_LENGTH (MSO_HEADER_LENGTH + 1)
00180 #define CHECK_VALIDATION_RESPONSE_CHECK_VALIDATION_STATUS_OFFSET
    (MSO_HEADER_LENGTH)
00181
00182 // Set Attribute Request
00183 // - Attribute id (1 byte)
00184 // - Index (1 byte)
00185 // - Value length (1 byte)
00186 // - Value (n bytes)
00187 #define SET_ATTRIBUTE_REQUEST_LENGTH 4
00188 #define SET_ATTRIBUTE_REQUEST_ATTRIBUTE_ID_OFFSET (MSO_HEADER_LENGTH)
00189 #define SET_ATTRIBUTE_REQUEST_INDEX_OFFSET (MSO_HEADER_LENGTH + 1)
00190 #define SET_ATTRIBUTE_REQUEST_VALUE_LENGTH_OFFSET (MSO_HEADER_LENGTH + 2)
00191 #define SET_ATTRIBUTE_REQUEST_VALUE_OFFSET (MSO_HEADER_LENGTH + 3)
00192
00193 // Set Attribute Response
00194 // - Attribute id (1 byte)
00195 // - Index (1 byte)
00196 // - Status (1 byte)
00197 #define SET_ATTRIBUTE_RESPONSE_LENGTH 4
00198 #define SET_ATTRIBUTE_RESPONSE_ATTRIBUTE_ID_OFFSET (MSO_HEADER_LENGTH)
00199 #define SET_ATTRIBUTE_RESPONSE_INDEX_OFFSET (MSO_HEADER_LENGTH + 1)
00200 #define SET_ATTRIBUTE_RESPONSE_STATUS_OFFSET (MSO_HEADER_LENGTH + 2)
00201
00202 // Get Attribute Request
00203 // - Attribute id (1 byte)
00204 // - Index (1 byte)
00205 // - Value length (1 byte)
00206 #define GET_ATTRIBUTE_REQUEST_LENGTH 4
00207 #define GET_ATTRIBUTE_REQUEST_ATTRIBUTE_ID_OFFSET (MSO_HEADER_LENGTH)
00208 #define GET_ATTRIBUTE_REQUEST_INDEX_OFFSET (MSO_HEADER_LENGTH + 1)
00209 #define GET_ATTRIBUTE_REQUEST_VALUE_LENGTH_OFFSET (MSO_HEADER_LENGTH + 2)
00210
00211 // Get Attribute Response
00212 // - Attribute id (1 byte)
00213 // - Index (1 byte)
00214 // - Status (1 byte)
00215 // - Value length (1 byte)
00216 // - Value (n bytes)
00217 #define GET_ATTRIBUTE_RESPONSE_LENGTH 5
00218 #define GET_ATTRIBUTE_RESPONSE_ATTRIBUTE_ID_OFFSET (MSO_HEADER_LENGTH)
00219 #define GET_ATTRIBUTE_RESPONSE_INDEX_OFFSET (MSO_HEADER_LENGTH + 1)
00220 #define GET_ATTRIBUTE_RESPONSE_STATUS_OFFSET (MSO_HEADER_LENGTH + 2)
00221 #define GET_ATTRIBUTE_RESPONSE_VALUE_LENGTH_OFFSET (MSO_HEADER_LENGTH + 3)
00222 #define GET_ATTRIBUTE_RESPONSE_VALUE_OFFSET (MSO_HEADER_LENGTH + 4)
00223
00224 // The User Control Pressed and User Control Repeated commands theoretically
00225 // take an unbounded additional payload, but the longest additional operand in
00226 // HDMI 1.3a is just four bytes. Still, just in case, leave an opening for the
00227 // user to override the buffer size.
00228 #ifndef EMBER_AF_PLUGIN_RF4CE_MSO_MAXIMUM_RC_COMMAND_PAYLOAD_LENGTH
00229     #define EMBER_AF_PLUGIN_RF4CE_MSO_MAXIMUM_RC_COMMAND_PAYLOAD_LENGTH 4
00230 #endif
00231 #define MAXIMUM_USER_CONTROL_X_LENGTH \
00232     (USER_CONTROL_PRESSED_LENGTH \
00233     + EMBER_AF_PLUGIN_RF4CE_MSO_MAXIMUM_RC_COMMAND_PAYLOAD_LENGTH)
00234
00235 // Assuming the standard operands are used, the Get Attribute Response command
00236 // is the commands with the longest payload in the MSO profile.
00237 #ifndef EMBER_AF_PLUGIN_RF4CE_MSO_MAXIMUM_PAYLOAD_LENGTH
00238     #if MAXIMUM_USER_CONTROL_X_LENGTH < GET_ATTRIBUTE_RESPONSE_LENGTH +
        APLC_MAX_RIB_ATTRIBUTE_SIZE
00239         #define EMBER_AF_PLUGIN_RF4CE_MSO_MAXIMUM_PAYLOAD_LENGTH
            (GET_ATTRIBUTE_RESPONSE_LENGTH + APLC_MAX_RIB_ATTRIBUTE_SIZE)
00240     #else
00241         #define EMBER_AF_PLUGIN_RF4CE_MSO_MAXIMUM_PAYLOAD_LENGTH
            MAXIMUM_USER_CONTROL_X_LENGTH
00242     #endif
00243 #endif
00244

```

```

00245 #define MSO_SET_VALUE_BINDING_RECIPIENT_PARAMETERS_BYTES_LENGTH
      5
00246
00247 // Internal binding state machine.
00248 enum {
00249     BINDING_STATE_GET_VALIDATION_CONFIG_PENDING
          = 0x00,
00250     BINDING_STATE_CHECK_VALIDATION_INITIAL
          = 0x01,
00251     BINDING_STATE_CHECK_VALIDATION_SENDING_REQUEST
          = 0x02,
00252     BINDING_STATE_CHECK_VALIDATION_IDLING
          = 0x03,
00253     BINDING_STATE_CHECK_VALIDATION_WAIT_FOR_RESPONSE
          = 0x04
00254 };
00255
00256 typedef struct {
00257     uint8_t channel;
00258     EmberPanId panId;
00259     EmberEUI64 ieeeAddr;
00260     uint8_t primaryClassDescriptor;
00261     uint8_t secondaryClassDescriptor;
00262     uint8_t tertiaryClassDescriptor;
00263     uint8_t basicLqiThreshold;
00264     uint8_t strictLqiThreshold;
00265     uint8_t rxLqi;
00266     uint8_t control;
00267 } EmAfMsoPairingCandidate;
00268
00269 extern uint8_t emAfRf4ceMsoBuffer[];
00270 extern uint8_t emAfRf4ceMsoBufferLength;
00271 extern EmberEventControl emberAfPluginRf4ceMsoUserControlEventControl
;
00272 extern EmberEventControl emberAfPluginRf4ceMsoCheckValidationEventControl
;
00273 extern EmberEventControl emberAfPluginRf4ceMsoSetGetAttributeEventControl
;
00274
00275 bool emAfRf4ceMsoIsBlackedOut(uint8_t pairingIndex);
00276 EmberStatus emAfRf4ceMsoSend(uint8_t pairingIndex);
00277 EmberStatus emAfRf4ceMsoSendExtended(uint8_t
pairingIndex,
                                         EmberRf4ceTxOption txOptions);
00278 void emAfRf4ceMsoMessageSent(uint8_t pairingIndex,
                                EmberAfRf4ceMsoCommandCode
commandCode,
                                const uint8_t *message,
                                uint8_t messageLength,
                                EmberStatus status);
00279 void emAfRf4ceMsoIncomingMessage(uint8_t
pairingIndex,
                                         EmberAfRf4ceMsoCommandCode
commandCode,
                                         const uint8_t *message,
                                         uint8_t messageLength);
00280
00281 void emAfRf4ceMsoSetValidation(uint8_t pairingIndex,
                                EmberAfRf4ceMsoValidationState
state,
                                EmberAfRf4ceMsoCheckValidationStatus
status);
00282 void emAfRf4ceMsoInitializeValidationStates
(void);
00283 EmberAfRf4ceMsoValidationState
emAfRf4ceMsoGetValidationState(uint8_t
pairingIndex);
00284 EmberAfRf4ceMsoCheckValidationStatus
emAfRf4ceMsoGetValidationStatus(uint8_t
pairingIndex);
00285 void emAfPluginRf4ceMsoCheckValidationRequestSentCallback
(EmberStatus status,
                                         uint8_t pairingIndex)
;
00286 void emAfPluginRf4ceMsoIncomingCheckValidationResponseCallback
(EmberAfRf4ceMsoCheckValidationStatus status
,
                                         uint8_t
pairingIndex);
00287 void emAfRf4ceMsoValidationConfigurationResponseCallback

```

```

        (EmberAfRf4ceStatus status);
00300 uint8_t emAfRf4ceMsoGetActiveBindPairingIndex
        (void);
00301 void emAfRf4ceMsoSetActiveBindPairingIndex
        (uint8_t pairingIndex);
00302
00303 EmberStatus emAfRf4ceMsoSetDiscoveryResponseUserString
        (void);
00304
00305 void emAfRf4ceMsoInitCommands(void);
00306
00307 #if defined(EMBER_SCRIPTED_TEST)
00308 #include "stack/core/ember-stack.h"
00309 #include "core/scripted-stub.h"
00310
00311 #define debugScriptCheck(reason)
        \
00312     simpleScriptCheck("scriptCheck", "scriptCheck: " reason, "")
00313 #else
00314 #define debugScriptCheck(reason)
00315 #endif // EMBER_SCRIPTED_TEST
00316
00317

```

8.30 rf4ce-mso-ir-rf-database-originator.h File Reference

Functions

- EmberStatus **emberAfRf4ceMsoIrRfDatabaseOriginatorGet** (EmberAfRf4ceMsoKeyCode keyCode, EmberAfRf4ceMsoIrRfDatabaseEntry *entry)
- EmberStatus **emberAfRf4ceMsoIrRfDatabaseOriginatorSet** (EmberAfRf4ceMsoKeyCode keyCode, const EmberAfRf4ceMsoIrRfDatabaseEntry *entry)
- EmberStatus **emberAfRf4ceMsoIrRfDatabaseOriginatorClear** (EmberAfRf4ceMsoKeyCode keyCode)
- void **emberAfRf4ceMsoIrRfDatabaseOriginatorClearAll** (void)

8.31 rf4ce-mso-ir-rf-database-originator.h

```

00001 // Copyright 2014 Silicon Laboratories, Inc.
00002
00003 #ifndef __RF4CE_MSO_IF_RF_DATABASE_ORIGINATOR_H__
00004 #define __RF4CE_MSO_IF_RF_DATABASE_ORIGINATOR_H__
00005
00035 #ifdef DOXYGEN_SHOULD_SKIP_THIS
00036
00045 EmberStatus emberAfRf4ceMsoIrRfDatabaseOriginatorGet
        (EmberAfRf4ceMsoKeyCode keyCode,
00046
        EmberAfRf4ceMsoIrRfDatabaseEntry *entry);
00047 #else
00048     #define emberAfRf4ceMsoIrRfDatabaseOriginatorGet \
00049         emberAfPluginRf4ceMsoGetIrRfDatabaseEntryCallback
00050 #endif
00051
00060 EmberStatus emberAfRf4ceMsoIrRfDatabaseOriginatorSet
        (EmberAfRf4ceMsoKeyCode keyCode,
00061
        const
        EmberAfRf4ceMsoIrRfDatabaseEntry *entry);
00062
00070 EmberStatus emberAfRf4ceMsoIrRfDatabaseOriginatorClear
        (EmberAfRf4ceMsoKeyCode keyCode);
00071
00073 void emberAfRf4ceMsoIrRfDatabaseOriginatorClearAll
        (void);
00074
00075 #endif /* __RF4CE_MSO_IF_RF_DATABASE_ORIGINATOR_H__ */
00076
00077 // END addtogroup

```

8.32 rf4ce-mso-ir-rf-database-recipient.h File Reference

Functions

- EmberStatus `emberAfRf4ceMsoIrRfDatabaseRecipientAdd` (`EmberAfRf4ceMsoKeyCode` keyCode, `EmberAfRf4ceMsoIrRfDatabaseEntry` *entry)
- EmberStatus `emberAfRf4ceMsoIrRfDatabaseRecipientRemove` (`EmberAfRf4ceMsoKeyCode` keyCode)
- void `emberAfRf4ceMsoIrRfDatabaseRecipientRemoveAll` (void)

8.33 rf4ce-mso-ir-rf-database-recipient.h

```

00001 // Copyright 2014 Silicon Laboratories, Inc.
00002
00003 #ifndef __RF4CE_MSO_IR_RF_DATABASE_RECIPIENT_H__
00004 #define __RF4CE_MSO_IR_RF_DATABASE_RECIPIENT_H__
00005
00046 EmberStatus emberAfRf4ceMsoIrRfDatabaseRecipientAdd
    (EmberAfRf4ceMsoKeyCode keyCode,
00047     EmberAfRf4ceMsoIrRfDatabaseEntry *entry);
00048
00049
00057 EmberStatus emberAfRf4ceMsoIrRfDatabaseRecipientRemove
    (EmberAfRf4ceMsoKeyCode keyCode);
00058
00059
00061 void emberAfRf4ceMsoIrRfDatabaseRecipientRemoveAll
    (void);
00062
00063
00064 #endif // __RF4CE_MSO_IR_RF_DATABASE_RECIPIENT_H__
00065
00066 // END addtogroup

```

8.34 rf4ce-mso-test.h File Reference

Macros

- `#define EMBER_AF_PLUGIN_RF4CE_MSO_KEY_REPEAT_INTERVAL_MS`
- `#define EMBER_AF_PLUGIN_RF4CE_MSO_MAX_USER_CONTROL_RECORDS`
- `#define EMBER_AF_PLUGIN_RF4CE_MSO_RESPONSE_WAIT_TIME_MS`
- `#define EMBER_AF_PLUGIN_RF4CE_MSO_AUTO_CHECK_VALIDATION_PERIOD_MS`
- `#define EMBER_AF_PLUGIN_RF4CE_MSO_LINK_LOST_WAIT_TIME_MS`
- `#define EMBER_AF_PLUGIN_RF4CE_MSO_MAX_PAIRING_CANDIDATES`
- `#define EMBER_AF_PLUGIN_RF4CE_MSO_KEY_EXCHANGE_TRANSFER_COUNT`
- `#define EMBER_AF_PLUGIN_RF4CE_PROFILE_VENDOR_ID`
- `#define EMBER_AF_PLUGIN_RF4CE_MSO_PERIPHERAL_ID_ENTRIES`
- `#define EMBER_AF_PLUGIN_RF4CE_MSO_GENERAL_PURPOSE_ENTRIES`

Functions

- void `emberAfPluginRf4ceMsoBindingCompleteCallback` (`EmberAfRf4ceMsoBindingStatus` status, `uint8_t` pairingIndex)
- void `emberAfPluginRf4ceMsoStartValidationCallback` (`uint8_t` pairingIndex)

- void `emberAfPluginRf4ceMsoUserControlCallback` (const `EmberAfRf4ceMsoUserControlRecord` *record)
- `EmberAfRf4ceStatus emberAfPluginRf4ceMsoGetIrRfDatabaseAttributeCallback` (uint8_t pairingIndex, uint8_t entryIndex, uint8_t *valueLength, uint8_t *value)
- bool `emberAfPluginRf4ceMsoHaveIrRfDatabaseAttributeCallback` (uint8_t pairingIndex, uint8_t entryIndex)
- void `emberAfPluginRf4ceMsoIncomingIrRfDatabaseAttributeCallback` (uint8_t pairingIndex, uint8_t entryIndex, uint8_t valueLength, const uint8_t *value)
- EmberStatus `emberAfPluginRf4ceMsoGetIrRfDatabaseEntryCallback` (`EmberAfRf4ceMsoKeyCode` keyCode, `EmberAfRf4ceMsoIrRfDatabaseEntry` *entry)

8.34.1 Macro Definition Documentation

8.34.1.1 #define EMBER_AF_PLUGIN_RF4CE_MSO_KEY_REPEAT_INTERVAL_MS

Definition at line 5 of file [rf4ce-mso-test.h](#).

8.34.1.2 #define EMBER_AF_PLUGIN_RF4CE_MSO_MAX_USER_CONTROL_RECORDS

Definition at line 6 of file [rf4ce-mso-test.h](#).

8.34.1.3 #define EMBER_AF_PLUGIN_RF4CE_MSO_RESPONSE_WAIT_TIME_MS

Definition at line 7 of file [rf4ce-mso-test.h](#).

8.34.1.4 #define EMBER_AF_PLUGIN_RF4CE_MSO_AUTO_CHECK_VALIDATION_PERIOD_MS

Definition at line 8 of file [rf4ce-mso-test.h](#).

8.34.1.5 #define EMBER_AF_PLUGIN_RF4CE_MSO_LINK_LOST_WAIT_TIME_MS

Definition at line 9 of file [rf4ce-mso-test.h](#).

8.34.1.6 #define EMBER_AF_PLUGIN_RF4CE_MSO_MAX_PAIRING_CANDIDATES

Definition at line 10 of file [rf4ce-mso-test.h](#).

8.34.1.7 #define EMBER_AF_PLUGIN_RF4CE_MSO_KEY_EXCHANGE_TRANSFER_COUNT

Definition at line 11 of file [rf4ce-mso-test.h](#).

8.34.1.8 #define EMBER_AF_PLUGIN_RF4CE_PROFILE_VENDOR_ID

Definition at line 12 of file [rf4ce-mso-test.h](#).

8.34.1.9 #define EMBER_AF_PLUGIN_RF4CE_MSO_PERIPHERAL_ID_ENTRIES

Definition at line 13 of file [rf4ce-mso-test.h](#).

8.34.1.10 #define EMBER_AF_PLUGIN_RF4CE_MSO_GENERAL_PURPOSE_ENTRIES

Definition at line 14 of file [rf4ce-mso-test.h](#).

8.35 rf4ce-mso-test.h

```

00001 // defines that are generated by app framework.
00002 // Forward declarations to avoid warnings in scripted tests (usually found in
00003 // the callback header file, that also generated by app framework).
00004
00005 #define EMBER_AF_PLUGIN_RF4CE_MSO_KEY_REPEAT_INTERVAL_MS           120
00006 #define EMBER_AF_PLUGIN_RF4CE_MSO_MAX_USER_CONTROL_RECORDS          1
00007 #define EMBER_AF_PLUGIN_RF4CE_MSO_RESPONSE_WAIT_TIME_MS             100
00008 #define EMBER_AF_PLUGIN_RF4CE_MSO_AUTO_CHECK_VALIDATION_PERIOD_MS    500
00009 #define EMBER_AF_PLUGIN_RF4CE_MSO_LINK_LOST_WAIT_TIME_MS            10000
00010 #define EMBER_AF_PLUGIN_RF4CE_MSO_MAX_PAIRING_CANDIDATES          3
00011 #define EMBER_AF_PLUGIN_RF4CE_MSO_KEY_EXCHANGE_TRANSFER_COUNT        4
00012 #define EMBER_AF_PLUGIN_RF4CE_PROFILE_VENDOR_ID                      0xABCD
00013 #define EMBER_AF_PLUGIN_RF4CE_MSO_PERIPHERAL_ID_ENTRIES              2
00014 #define EMBER_AF_PLUGIN_RF4CE_MSO_GENERAL_PURPOSE_ENTRIES          1
00015
00016 void emberAfPluginRf4ceMsoBindingCompleteCallback
  (EmberAfRf4ceMsoBindingStatus status,
   uint8_t pairingIndex);
00017
00018
00019 void emberAfPluginRf4ceMsoStartValidationCallback
  (uint8_t pairingIndex);
00020
00021 void emberAfPluginRf4ceMsoUserControlCallback
  (const EmberAfRf4ceMsoUserControlRecord *record)
;
00022
00023 EmberAfRf4ceStatus
emberAfPluginRf4ceMsoGetIrRfDatabaseAttributeCallback
  (uint8_t pairingIndex,
00024   uint8_t entryIndex,
00025   uint8_t *valueLength,
00026   uint8_t *value);
00027
00028 bool emberAfPluginRf4ceMsoHaveIrRfDatabaseAttributeCallback
  (uint8_t pairingIndex,
   uint8_t
00029   entryIndex);
00030
00031 void emberAfPluginRf4ceMsoIncomingIrRfDatabaseAttributeCallback
  (uint8_t pairingIndex,
   uint8_t
00032   entryIndex,
   uint8_t
00033   valueLength,
   uint8_t
00034   const uint8_t *
   value);
00035 EmberStatus emberAfPluginRf4ceMsoGetIrRfDatabaseEntryCallback
  (EmberAfRf4ceMsoKeyCode keyCode,
00036   EmberAfRf4ceMsoIrRfDatabaseEntry *entry);

```

8.36 rf4ce-mso-tokens.h File Reference

8.37 rf4ce-mso-tokens.h

```

00001 // Copyright 2013 Silicon Laboratories, Inc.
00002
00003 #ifdef EMBER_AF_RF4CE_NODE_TYPE_CONTROLLER
00004     // This token stores the pairing index of the current active bind, if any.
00005     #define CREATOR_PLUGIN_RF4CE_MSO_ACTIVE_BIND_PAIRING_INDEX 0x8726
00006     #ifndef DEFINETOKENS
00007         DEFINE_BASIC_TOKEN(PLUGIN_RF4CE_MSO_ACTIVE_BIND_PAIRING_INDEX,
00008             uint8_t,
00009             0xFF)
00010     #endif
00011 #endif
00012
00013 #ifdef EMBER_AF_RF4CE_NODE_TYPE_TARGET
00014     #define CREATOR_PLUGIN_RF4CE_MSO_VALIDATION_TABLE 0x8727
00015     #ifndef DEFINETYPES
00016         #include "rf4ce-mso-types.h"
00017         typedef struct {
00018             EmberAfRf4ceMsoValidationState state;
00019             EmberAfRf4ceMsoCheckValidationStatus
00020             status;
00021         } tokTypePluginRf4ceMsoValidation;
00022     #endif
00023     #ifndef DEFINETOKENS
00024         DEFINE_INDEXED_TOKEN(PLUGIN_RF4CE_MSO_VALIDATION_TABLE,
00025             tokTypePluginRf4ceMsoValidation,
00026             EMBER_RF4CE_PAIRING_TABLE_SIZE,
00027             {EMBER_AF_RF4CE_MSO_VALIDATION_STATE_NOT_VALIDATED
00028             ,
00029             EMBER_AF_RF4CE_MSO_CHECK_VALIDATION_STATUS_FAILURE
00030         })
00031     #endif
00032     #define CREATOR_PLUGIN_RF4CE_MSO_ATTRIBUTE_PERIPHERAL_IDS
00033     x8728
00034     #define CREATOR_PLUGIN_RF4CE_MSO_ATTRIBUTE_RF_STATISTICS
00035     x8729
00036     #define CREATOR_PLUGIN_RF4CE_MSO_ATTRIBUTE_VERSIONING
00037     x8730
00038     #define CREATOR_PLUGIN_RF4CE_MSO_ATTRIBUTE_BATTERY_STATUS
00039     x8731
00040     #define CREATOR_PLUGIN_RF4CE_MSO_ATTRIBUTE_SHOR_RF_RETRY_PERIOD
00041     x8732
00042     #define CREATOR_PLUGIN_RF4CE_MSO_ATTRIBUTE_VALIDATION_CONFIGURATION
00043     x8733
00044     #define CREATOR_PLUGIN_RF4CE_MSO_ATTRIBUTE_GENERAL_PURPOSE
00045     x8734
00046
00047     #ifndef DEFINETYPES
00048         #include "rf4ce-mso-types.h"
00049         #include "rf4ce-mso-attributes.h"
00050         typedef EmAfRf4ceMsoPeripheralIdEntry
00051         tokMsoAttributePeripheralIds[EMBER_AF_PLUGIN_RF4CE_MSO_PERIPHERAL_ID_ENTRIES
00052         ];
00053         typedef uint8_t tokMsoAttributeRfStatistics[
00054             MSO_RIB_ATTRIBUTE_RF_STATISTICS_LENGTH];
00055         typedef uint8_t tokMsoAttributeVersioning[MSO_ATTRIBUTE_VERSIONING_ENTRIES
00056             ][MSO_RIB_ATTRIBUTE_VERSIONING_LENGTH];
00057         typedef uint8_t tokMsoAttributeBatteryStatus[
00058             MSO_RIB_ATTRIBUTE_BATTERY_STATUS_LENGTH]
00059         ;
00060         typedef uint8_t tokMsoAttributeShortRfRetryPeriod[
00061             MSO_RIB_ATTRIBUTE_SHORT_RF_RETRY_PERIOD_LENGTH
00062         ];
00063         typedef uint8_t tokMsoAttributeValidationConfiguration[
00064             MSO_RIB_ATTRIBUTE_VALIDATION_CONFIGURATION_LENGTH
00065         ];
00066         typedef uint8_t tokMsoAttributeGeneralPurpose[
00067             EMBER_AF_PLUGIN_RF4CE_MSO_GENERAL_PURPOSE_ENTRIES
00068             ][MSO_RIB_ATTRIBUTE_GENERAL_PURPOSE_LENGTH
00069         ];
00070     #endif
00071     #ifndef DEFINETOKENS
00072         DEFINE_INDEXED_TOKEN(PLUGIN_RF4CE_MSO_ATTRIBUTE_PERIPHERAL_IDS,
00073             tokMsoAttributePeripheralIds,
00074             EMBER_RF4CE_PAIRING_TABLE_SIZE,
00075             {{0xFF, {0xFF, 0xFF, 0xFF, 0xFF}}})
00076         DEFINE_INDEXED_TOKEN(PLUGIN_RF4CE_MSO_ATTRIBUTE_RF_STATISTICS,

```

```

00055             tokMsoAttributeRfStatistics,
00056             EMBER_RF4CE_PAIRING_TABLE_SIZE,
00057             {0,})
00058     DEFINE_INDEXED_TOKEN(PLUGIN_RF4CE_MSO_ATTRIBUTE_VERSIONING,
00059                         tokMsoAttributeVersioning,
00060                         EMBER_RF4CE_PAIRING_TABLE_SIZE,
00061                         {0,})
00062     DEFINE_INDEXED_TOKEN(PLUGIN_RF4CE_MSO_ATTRIBUTE_BATTERY_STATUS,
00063                         tokMsoAttributeBatteryStatus,
00064                         EMBER_RF4CE_PAIRING_TABLE_SIZE,
00065                         {0,})
00066     DEFINE_INDEXED_TOKEN(PLUGIN_RF4CE_MSO_ATTRIBUTE_SHOR_RF_RETRY_PERIOD,
00067                         tokMsoAttributeShortRfRetryPeriod,
00068                         EMBER_RF4CE_PAIRING_TABLE_SIZE,
00069                         {0,})
00070     DEFINE_INDEXED_TOKEN(PLUGIN_RF4CE_MSO_ATTRIBUTE_VALIDATION_CONFIGURATION,
00071                         tokMsoAttributeValidationConfiguration,
00072                         EMBER_RF4CE_PAIRING_TABLE_SIZE,
00073                         {LOW_BYTE(
00074                             EMBER_AF_PLUGIN_RF4CE_MSO_LINK_LOST_WAIT_TIME_MS
00075                         ),
00076                             HIGH_BYTE(
00077                             EMBER_AF_PLUGIN_RF4CE_MSO_LINK_LOST_WAIT_TIME_MS
00078                         ),
00079                             LOW_BYTE(
00080                             EMBER_AF_PLUGIN_RF4CE_MSO_AUTO_CHECK_VALIDATION_PERIOD_MS
00081                         ),
00082                             HIGH_BYTE(
00083                             EMBER_AF_PLUGIN_RF4CE_MSO_AUTO_CHECK_VALIDATION_PERIOD_MS
00084                         )));
00085     DEFINE_INDEXED_TOKEN(PLUGIN_RF4CE_MSO_ATTRIBUTE_GENERAL_PURPOSE,
00086                         tokMsoAttributeGeneralPurpose,
00087                         EMBER_RF4CE_PAIRING_TABLE_SIZE,
00088                         {0,})
00089 #endif
00090#endif

```

8.38 rf4ce-mso-types.h File Reference

Data Structures

- struct [EmberAfRf4ceMsoUserControlRecord](#)
This data structure contains the MSO user control record.
- struct [EmberAfRf4ceMsoIrRfDatabaseRfDescriptor](#)
RF4CE MSO IR-RF database RF descriptor.
- struct [EmberAfRf4ceMsoIrRfDatabaseIrDescriptor](#)
RF4CE MSO IR-RF database IR descriptor.
- struct [EmberAfRf4ceMsoIrRfDatabaseEntry](#)
RF4CE MSO IR-RF database entry.

Enumerations

- enum [EmberAfRf4ceMsoBindingState](#) { EMBER_AF_RF4CE_MSO_BINDING_STATE_NOT_BOUND, EMBER_AF_RF4CE_MSO_BINDING_STATE_BINDING, EMBER_AF_RF4CE_MSO_BINDING_STATE_BOUND }
- enum [EmberAfRf4ceMsoValidationState](#) { EMBER_AF_RF4CE_MSO_VALIDATION_STATE_NOT_VALIDATED, EMBER_AF_RF4CE_MSO_VALIDATION_STATE_REJECTED, EMBER_AF_RF4CE_MSO_VALIDATION_STATE_VALIDATING, EMBER_AF_RF4CE_MSO_VALIDATION_STATE_REVALIDATING, EMBER_AF_RF4CE_MSO_VALIDATION_STATE_VALIDATED }

- enum EmberAfRf4ceMsoCommandCode {
 EMBER_AF_RF4CE_MSO_COMMAND_USER_CONTROL_PRESSED, EMBER_AF_RF4CE_MSO_COMMAND_USER_CONTROL_REPEATED, EMBER_AF_RF4CE_MSO_COMMAND_USER_CONTROL_RELEASED, EMBER_AF_RF4CE_MSO_COMMAND_CHECK_VALIDATION_REQUEST,
 EMBER_AF_RF4CE_MSO_COMMAND_CHECK_VALIDATION_RESPONSE, EMBER_AF_RF4CE_MSO_COMMAND_SET_ATTRIBUTE_REQUEST, EMBER_AF_RF4CE_MSO_COMMAND_SET_ATTRIBUTE_RESPONSE, EMBER_AF_RF4CE_MSO_COMMAND_GET_ATTRIBUTE_REQUEST,
 EMBER_AF_RF4CE_MSO_COMMAND_GET_ATTRIBUTE_RESPONSE }

- enum EmberAfRf4ceMsoKeyCode {
 EMBER_AF_RF4CE_MSO_KEY_CODE_OK, EMBER_AF_RF4CE_MSO_KEY_CODE_UP_ARROW, EMBER_AF_RF4CE_MSO_KEY_CODE_DOWN_ARROW, EMBER_AF_RF4CE_MSO_KEY_CODE_LEFT_ARROW, EMBER_AF_RF4CE_MSO_KEY_CODE_RIGHT_ARROW, EMBER_AF_RF4CE_MSO_KEY_CODE_MENU, EMBER_AF_RF4CE_MSO_KEY_CODE_DVR, EMBER_AF_RF4CE_MSO_KEY_CODE_FAV, EMBER_AF_RF4CE_MSO_KEY_CODE_EXIT, EMBER_AF_RF4CE_MSO_KEY_CODE_HOME, EMBER_AF_RF4CE_MSO_KEY_CODE_DIGIT_0, EMBER_AF_RF4CE_MSO_KEY_CODE_DIGIT_1, EMBER_AF_RF4CE_MSO_KEY_CODE_DIGIT_2, EMBER_AF_RF4CE_MSO_KEY_CODE_DIGIT_3, EMBER_AF_RF4CE_MSO_KEY_CODE_DIGIT_4, EMBER_AF_RF4CE_MSO_KEY_CODE_DIGIT_5, EMBER_AF_RF4CE_MSO_KEY_CODE_DIGIT_6, EMBER_AF_RF4CE_MSO_KEY_CODE_DIGIT_7, EMBER_AF_RF4CE_MSO_KEY_CODE_DIGIT_8, EMBER_AF_RF4CE_MSO_KEY_CODE_DIGIT_9, EMBER_AF_RF4CE_MSO_KEY_CODE_FULL_STOP, EMBER_AF_RF4CE_MSO_KEY_CODE_RETURN, EMBER_AF_RF4CE_MSO_KEY_CODE_CHANNEL_UP, EMBER_AF_RF4CE_MSO_KEY_CODE_CHANNEL_DOWN, EMBER_AF_RF4CE_MSO_KEY_CODE_LAST, EMBER_AF_RF4CE_MSO_KEY_CODE_INPUT_SELECT, EMBER_AF_RF4CE_MSO_KEY_CODE_INFO, EMBER_AF_RF4CE_MSO_KEY_CODE_HELP, EMBER_AF_RF4CE_MSO_KEY_CODE_PAGE_UP, EMBER_AF_RF4CE_MSO_KEY_CODE_PAGE_DOWN, EMBER_AF_RF4CE_MSO_KEY_CODE_MOTION, EMBER_AF_RF4CE_MSO_KEY_CODE_SEARCH, EMBER_AF_RF4CE_MSO_KEY_CODE_LIVE, EMBER_AF_RF4CE_MSO_KEY_CODE_HD_ZOOM, EMBER_AF_RF4CE_MSO_KEY_CODE_SHARE, EMBER_AF_RF4CE_MSO_KEY_CODE_TV_POWER, EMBER_AF_RF4CE_MSO_KEY_CODE_VOLUME_UP, EMBER_AF_RF4CE_MSO_KEY_CODE_VOLUME_DOWN, EMBER_AF_RF4CE_MSO_KEY_CODE_MUTE, EMBER_AF_RF4CE_MSO_KEY_CODE_PLAY, EMBER_AF_RF4CE_MSO_KEY_CODE_STOP, EMBER_AF_RF4CE_MSO_KEY_CODE_PAUSE, EMBER_AF_RF4CE_MSO_KEY_CODE_RECORD, EMBER_AF_RF4CE_MSO_KEY_CODE_REWIND, EMBER_AF_RF4CE_MSO_KEY_CODE_FAST_FORWARD, EMBER_AF_RF4CE_MSO_KEY_CODE_30_SECOND_SKIP_AHEAD, EMBER_AF_RF4CE_MSO_KEY_CODE_REPLAY, EMBER_AF_RF4CE_MSO_KEY_CODE_SWAP, EMBER_AF_RF4CE_MSO_KEY_CODE_ON_DEMAND, EMBER_AF_RF4CE_MSO_KEY_CODE_GUIDE, EMBER_AF_RF4CE_MSO_KEY_CODE_PUSH_TO_TALK, EMBER_AF_RF4CE_MSO_KEY_CODE_PIP_ON_OFF, EMBER_AF_RF4CE_MSO_KEY_CODE_PIP_MOVE, EMBER_AF_RF4CE_MSO_KEY_CODE_PIP_CHANNEL_UP, EMBER_AF_RF4CE_MSO_KEY_CODE_PIP_CHANNEL_DOWN, EMBER_AF_RF4CE_MSO_KEY_CODE_LOCK, EMBER_AF_RF4CE_MSO_KEY_CODE_DAY_UP, EMBER_AF_RF4CE_MSO_KEY_CODE_DAY_DOWN, EMBER_AF_RF4CE_MSO_KEY_CODE_PLAY_PAUSE, EMBER_AF_RF4CE_MSO_KEY_CODE_STOP_VIDEO, EMBER_AF_RF4CE_MSO_KEY_CODE_MUTE_MIC, EMBER_AF_RF4CE_MSO_KEY_CODE_POWER_TOGGLE, EMBER_AF_RF4CE_MSO_KEY_CODE_POWER_OFF, EMBER_AF_RF4CE_MSO_KEY_CODE_POWER_ON, EMBER_AF_RF4CE_MSO_KEY_CODE_OCAP_B, EMBER_AF_RF4CE_MSO_KEY_CODE_BLUE_SQUARE, EMBER_AF_RF4CE_MSO_KEY_CODE_OCAP_C, EMBER_AF_RF4CE_MSO_KEY_CODE_RED_CIRCLE, EMBER_AF_RF4CE_MSO_KEY_CODE_OCAP_D, EMBER_AF_RF4CE_MSO_KEY_CODE_GREEN_DIAMOND, EMBER_AF_RF4CE_MSO_KEY_CODE_OCAP_A, EMBER_AF_RF4CE_MSO_KEY_CODE_YELLOW_TRIANGLE, EMBER_AF_RF4CE_MSO_KEY_CODE_PROFILE, EMBER_AF_RF4CE_MSO_KEY_CODE_CALL, EMBER_AF_RF4CE_MSO_KEY_CODE_HOLD

- enum `EmberAfRf4ceMsoCheckValidationControl` { `EMBER_AF_RF4CE_MSO_CHECK_VALIDATION_CONTROL_REQUEST_AUTOMATIC_VALIDATION` }
- enum `EmberAfRf4ceMsoCheckValidationStatus` {
 `EMBER_AF_RF4CE_MSO_CHECK_VALIDATION_STATUS_SUCCESS`, `EMBER_AF_RF4CE_MSO_CHECK_VALIDATION_STATUS_PENDING`, `EMBER_AF_RF4CE_MSO_CHECK_VALIDATION_STATUS_TIMEOUT`, `EMBER_AF_RF4CE_MSO_CHECK_VALIDATION_STATUS_COLLISION`,
 `EMBER_AF_RF4CE_MSO_CHECK_VALIDATION_STATUS_FAILURE`, `EMBER_AF_RF4CE_MSO_CHECK_VALIDATION_STATUS_ABORT`, `EMBER_AF_RF4CE_MSO_CHECK_VALIDATION_STATUS_FULL_ABORT` }
- enum `EmberAfRf4ceMsoBindingStatus` {
 `EMBER_AF_RF4CE_MSO_BINDING_STATUS_SUCCESS`, `EMBER_AF_RF4CE_MSO_BINDING_STATUS_NO_VALID_RESPONSE`, `EMBER_AF_RF4CE_MSO_BINDING_STATUS_NO_VALID_CANDIDATE`, `EMBER_AF_RF4CE_MSO_BINDING_STATUS_DUPLICATE_CLASS_ABORT`,
 `EMBER_AF_RF4CE_MSO_BINDING_STATUS_PAIRING_FAILED`, `EMBER_AF_RF4CE_MSO_BINDING_STATUS_VALIDATION_TIMEOUT`, `EMBER_AF_RF4CE_MSO_BINDING_STATUS_VALIDATION_COLLISION`, `EMBER_AF_RF4CE_MSO_BINDING_STATUS_VALIDATION_FAILURE`,
 `EMBER_AF_RF4CE_MSO_BINDING_STATUS_VALIDATION_ABORT`, `EMBER_AF_RF4CE_MSO_BINDING_STATUS_VALIDATION_FULL_ABORT` }
- enum `EmberAfRf4ceMsoAttributeId` {
 `EMBER_AF_RF4CE_MSO_ATTRIBUTE_ID_PERIPHERAL_IDS`, `EMBER_AF_RF4CE_MSO_ATTRIBUTE_ID_RF_STATISTICS`, `EMBER_AF_RF4CE_MSO_ATTRIBUTE_ID_VERSIONING`, `EMBER_AF_RF4CE_MSO_ATTRIBUTE_ID_BATTERY_STATUS`,
 `EMBER_AF_RF4CE_MSO_ATTRIBUTE_ID_SHORT_RF_RETRY_PERIOD`, `EMBER_AF_RF4CE_MSO_ATTRIBUTE_ID_VALIDATION_CONFIGURATION`, `EMBER_AF_RF4CE_MSO_ATTRIBUTE_ID_GENERAL_PURPOSE` }
- enum `EmberAfRf4ceMsoIrRfDatabaseFlags` {
 `EMBER_AF_RF4CE_MSO_IR_RF_DATABASE_FLAG_NONE`, `EMBER_AF_RF4CE_MSO_IR_RF_DATABASE_FLAG_RF_PRESSED_SPECIFIED`, `EMBER_AF_RF4CE_MSO_IR_RF_DATABASE_FLAG_RF_REPEAT_SPECIFIED`, `EMBER_AF_RF4CE_MSO_IR_RF_DATABASE_FLAG_RF_RELEASED_SPECIFIED`,
 `EMBER_AF_RF4CE_MSO_IR_RF_DATABASE_FLAG_IR_SPECIFIED`, `EMBER_AF_RF4CE_MSO_IR_RF_DATABASE_FLAG_DEVICE_TYPE_MASK`, `EMBER_AF_RF4CE_MSO_IR_RF_DATABASE_FLAG_USE_DEFAULT`, `EMBER_AF_RF4CE_MSO_IR_RF_DATABASE_FLAG_PERMANENT` }
- enum `EmberAfRf4ceMsoIrRfDatabaseDeviceType` { `EMBER_AF_RF4CE_MSO_IR_RF_DATABASE_DEVICE_TYPE_TV`, `EMBER_AF_RF4CE_MSO_IR_RF_DATABASE_DEVICE_TYPE_AVR` }
- enum `EmberAfRf4ceMsoIrRfDatabaseRfConfig` { `EMBER_AF_RF4CE_MSO_IR_RF_DATABASE_RF_CONFIG_MINIMUM_NUMBER_OF_TRANSMISSIONS_MASK`, `EMBER_AF_RF4CE_MSO_IR_RF_DATABASE_RF_CONFIG_KEEP_TRANSMITTING_UNTIL_KEY_RELEASE`, `EMBER_AF_RF4CE_MSO_IR_RF_DATABASE_RF_CONFIG_SHORT_RETRY` }
- enum `EmberAfRf4ceMsoIrRfDatabaseIrConfig` { `EMBER_AF_RF4CE_MSO_IR_RF_DATABASE_IR_CONFIG_MINIMUM_NUMBER_OF_TRANSMISSIONS_MASK`, `EMBER_AF_RF4CE_MSO_IR_RF_DATABASE_IR_CONFIG_KEEP_TRANSMITTING_UNTIL_KEY_RELEASE`, `EMBER_AF_RF4CE_MSO_IR_RF_DATABASE_IR_CONFIG_SHORT_RETRY` }

8.38.1 Enumeration Type Documentation

8.38.1.1 enum EmberAfRf4ceMsoBindingState

RF4CE MSO binding states.

Enumerator:

EMBER_AF_RF4CE_MSO_BINDING_STATE_NOT_BOUND
EMBER_AF_RF4CE_MSO_BINDING_STATE_BINDING
EMBER_AF_RF4CE_MSO_BINDING_STATE_BOUND

Definition at line 10 of file [rf4ce-mso-types.h](#).

8.38.1.2 enum EmberAfRf4ceMsoValidationState

RF4CE MSO validation states.

Enumerator:

EMBER_AF_RF4CE_MSO_VALIDATION_STATE_NOT_VALIDATED
EMBER_AF_RF4CE_MSO_VALIDATION_STATE_REJECTED
EMBER_AF_RF4CE_MSO_VALIDATION_STATE_VALIDATING
EMBER_AF_RF4CE_MSO_VALIDATION_STATE_REVALIDATING
EMBER_AF_RF4CE_MSO_VALIDATION_STATE_VALIDATED

Definition at line 25 of file [rf4ce-mso-types.h](#).

8.38.1.3 enum EmberAfRf4ceMsoCommandCode

RF4CE MSO command codes.

Enumerator:

EMBER_AF_RF4CE_MSO_COMMAND_USER_CONTROL_PRESSED
EMBER_AF_RF4CE_MSO_COMMAND_USER_CONTROL_REPEAT
EMBER_AF_RF4CE_MSO_COMMAND_USER_CONTROL_RELEASED
EMBER_AF_RF4CE_MSO_COMMAND_CHECK_VALIDATION_REQUEST
EMBER_AF_RF4CE_MSO_COMMAND_CHECK_VALIDATION_RESPONSE
EMBER_AF_RF4CE_MSO_COMMAND_SET_ATTRIBUTE_REQUEST
EMBER_AF_RF4CE_MSO_COMMAND_SET_ATTRIBUTE_RESPONSE
EMBER_AF_RF4CE_MSO_COMMAND_GET_ATTRIBUTE_REQUEST
EMBER_AF_RF4CE_MSO_COMMAND_GET_ATTRIBUTE_RESPONSE

Definition at line 42 of file [rf4ce-mso-types.h](#).

8.38.1.4 enum EmberAfRf4ceMsoKeyCode

RF4CE MSO key codes.

Enumerator:

```
EMBER_AF_RF4CE_MSO_KEY_CODE_OK
EMBER_AF_RF4CE_MSO_KEY_CODE_UP_ARROW
EMBER_AF_RF4CE_MSO_KEY_CODE_DOWN_ARROW
EMBER_AF_RF4CE_MSO_KEY_CODE_LEFT_ARROW
EMBER_AF_RF4CE_MSO_KEY_CODE_RIGHT_ARROW
EMBER_AF_RF4CE_MSO_KEY_CODE_MENU
EMBER_AF_RF4CE_MSO_KEY_CODE_DVR
EMBER_AF_RF4CE_MSO_KEY_CODE_FAV
EMBER_AF_RF4CE_MSO_KEY_CODE_EXIT
EMBER_AF_RF4CE_MSO_KEY_CODE_HOME
EMBER_AF_RF4CE_MSO_KEY_CODE_DIGIT_0
EMBER_AF_RF4CE_MSO_KEY_CODE_DIGIT_1
EMBER_AF_RF4CE_MSO_KEY_CODE_DIGIT_2
EMBER_AF_RF4CE_MSO_KEY_CODE_DIGIT_3
EMBER_AF_RF4CE_MSO_KEY_CODE_DIGIT_4
EMBER_AF_RF4CE_MSO_KEY_CODE_DIGIT_5
EMBER_AF_RF4CE_MSO_KEY_CODE_DIGIT_6
EMBER_AF_RF4CE_MSO_KEY_CODE_DIGIT_7
EMBER_AF_RF4CE_MSO_KEY_CODE_DIGIT_8
EMBER_AF_RF4CE_MSO_KEY_CODE_DIGIT_9
EMBER_AF_RF4CE_MSO_KEY_CODE_FULL_STOP
EMBER_AF_RF4CE_MSO_KEY_CODE_RETURN
EMBER_AF_RF4CE_MSO_KEY_CODE_CHANNEL_UP
EMBER_AF_RF4CE_MSO_KEY_CODE_CHANNEL_DOWN
EMBER_AF_RF4CE_MSO_KEY_CODE_LAST
EMBER_AF_RF4CE_MSO_KEY_CODE_LANG
EMBER_AF_RF4CE_MSO_KEY_CODE_INPUT_SELECT
EMBER_AF_RF4CE_MSO_KEY_CODE_INFO
EMBER_AF_RF4CE_MSO_KEY_CODE_HELP
EMBER_AF_RF4CE_MSO_KEY_CODE_PAGE_UP
EMBER_AF_RF4CE_MSO_KEY_CODE_PAGE_DOWN
EMBER_AF_RF4CE_MSO_KEY_CODE_MOTION
EMBER_AF_RF4CE_MSO_KEY_CODE_SEARCH
EMBER_AF_RF4CE_MSO_KEY_CODE_LIVE
EMBER_AF_RF4CE_MSO_KEY_CODE_HD_ZOOM
EMBER_AF_RF4CE_MSO_KEY_CODE_SHARE
EMBER_AF_RF4CE_MSO_KEY_CODE_TV_POWER
```

*EMBER_AF_RF4CE_MSO_KEY_CODE_VOLUME_UP
EMBER_AF_RF4CE_MSO_KEY_CODE_VOLUME_DOWN
EMBER_AF_RF4CE_MSO_KEY_CODE_MUTE
EMBER_AF_RF4CE_MSO_KEY_CODE_PLAY
EMBER_AF_RF4CE_MSO_KEY_CODE_STOP
EMBER_AF_RF4CE_MSO_KEY_CODE_PAUSE
EMBER_AF_RF4CE_MSO_KEY_CODE_RECORD
EMBER_AF_RF4CE_MSO_KEY_CODE_REWIND
EMBER_AF_RF4CE_MSO_KEY_CODE_FAST_FORWARD
EMBER_AF_RF4CE_MSO_KEY_CODE_30_SECOND_SKIP_AHEAD
EMBER_AF_RF4CE_MSO_KEY_CODE_REPEAT
EMBER_AF_RF4CE_MSO_KEY_CODE_SWAP
EMBER_AF_RF4CE_MSO_KEY_CODE_ON_DEMAND
EMBER_AF_RF4CE_MSO_KEY_CODE_GUIDE
EMBER_AF_RF4CE_MSO_KEY_CODE_PUSH_TO_TALK
EMBER_AF_RF4CE_MSO_KEY_CODE_PIP_ON_OFF
EMBER_AF_RF4CE_MSO_KEY_CODE_PIP_MOVE
EMBER_AF_RF4CE_MSO_KEY_CODE_PIP_CHANNEL_UP
EMBER_AF_RF4CE_MSO_KEY_CODE_PIP_CHANNEL_DOWN
EMBER_AF_RF4CE_MSO_KEY_CODE_LOCK
EMBER_AF_RF4CE_MSO_KEY_CODE_DAY_UP
EMBER_AF_RF4CE_MSO_KEY_CODE_DAY_DOWN
EMBER_AF_RF4CE_MSO_KEY_CODE_PLAY_PAUSE
EMBER_AF_RF4CE_MSO_KEY_CODE_STOP_VIDEO
EMBER_AF_RF4CE_MSO_KEY_CODE_MUTE_MIC
EMBER_AF_RF4CE_MSO_KEY_CODE_POWER_TOGGLE
EMBER_AF_RF4CE_MSO_KEY_CODE_POWER_OFF
EMBER_AF_RF4CE_MSO_KEY_CODE_POWER_ON
EMBER_AF_RF4CE_MSO_KEY_CODE_OCAP_B
EMBER_AF_RF4CE_MSO_KEY_CODE_BLUE_SQUARE
EMBER_AF_RF4CE_MSO_KEY_CODE_OCAP_C
EMBER_AF_RF4CE_MSO_KEY_CODE_RED_CIRCLE
EMBER_AF_RF4CE_MSO_KEY_CODE_OCAP_D
EMBER_AF_RF4CE_MSO_KEY_CODE_GREEN_DIAMOND
EMBER_AF_RF4CE_MSO_KEY_CODE_OCAP_A
EMBER_AF_RF4CE_MSO_KEY_CODE_YELLOW_TRIANGLE
EMBER_AF_RF4CE_MSO_KEY_CODE_PROFILE
EMBER_AF_RF4CE_MSO_KEY_CODE_CALL
EMBER_AF_RF4CE_MSO_KEY_CODE_HOLD
EMBER_AF_RF4CE_MSO_KEY_CODE_END
EMBER_AF_RF4CE_MSO_KEY_CODE_VIEWS*

*EMBER_AF_RF4CE_MS0_KEY_CODE_SELF_VIEW
EMBER_AF_RF4CE_MS0_KEY_CODE_ZOOM_IN
EMBER_AF_RF4CE_MS0_KEY_CODE_ZOOM_OUT
EMBER_AF_RF4CE_MS0_KEY_CODE_BACKSPACE
EMBER_AF_RF4CE_MS0_KEY_CODE_LOCK_UNLOCK
EMBER_AF_RF4CE_MS0_KEY_CODE_CAPS
EMBER_AF_RF4CE_MS0_KEY_CODE_ALT
EMBER_AF_RF4CE_MS0_KEY_CODE_SPACE
EMBER_AF_RF4CE_MS0_KEY_CODE_WWW_DOT
EMBER_AF_RF4CE_MS0_KEY_CODE_DOT_COM
EMBER_AF_RF4CE_MS0_KEY_CODE_LATIN_CAPITAL_LETTER_A
EMBER_AF_RF4CE_MS0_KEY_CODE_LATIN_CAPITAL_LETTER_B
EMBER_AF_RF4CE_MS0_KEY_CODE_LATIN_CAPITAL_LETTER_C
EMBER_AF_RF4CE_MS0_KEY_CODE_LATIN_CAPITAL_LETTER_D
EMBER_AF_RF4CE_MS0_KEY_CODE_LATIN_CAPITAL_LETTER_E
EMBER_AF_RF4CE_MS0_KEY_CODE_LATIN_CAPITAL_LETTER_F
EMBER_AF_RF4CE_MS0_KEY_CODE_LATIN_CAPITAL_LETTER_G
EMBER_AF_RF4CE_MS0_KEY_CODE_LATIN_CAPITAL_LETTER_H
EMBER_AF_RF4CE_MS0_KEY_CODE_LATIN_CAPITAL_LETTER_I
EMBER_AF_RF4CE_MS0_KEY_CODE_LATIN_CAPITAL_LETTER_J
EMBER_AF_RF4CE_MS0_KEY_CODE_LATIN_CAPITAL_LETTER_K
EMBER_AF_RF4CE_MS0_KEY_CODE_LATIN_CAPITAL_LETTER_L
EMBER_AF_RF4CE_MS0_KEY_CODE_LATIN_CAPITAL_LETTER_M
EMBER_AF_RF4CE_MS0_KEY_CODE_LATIN_CAPITAL_LETTER_N
EMBER_AF_RF4CE_MS0_KEY_CODE_LATIN_CAPITAL_LETTER_O
EMBER_AF_RF4CE_MS0_KEY_CODE_LATIN_CAPITAL_LETTER_P
EMBER_AF_RF4CE_MS0_KEY_CODE_LATIN_CAPITAL_LETTER_Q
EMBER_AF_RF4CE_MS0_KEY_CODE_LATIN_CAPITAL_LETTER_R
EMBER_AF_RF4CE_MS0_KEY_CODE_LATIN_CAPITAL_LETTER_S
EMBER_AF_RF4CE_MS0_KEY_CODE_LATIN_CAPITAL_LETTER_T
EMBER_AF_RF4CE_MS0_KEY_CODE_LATIN_CAPITAL_LETTER_U
EMBER_AF_RF4CE_MS0_KEY_CODE_LATIN_CAPITAL_LETTER_V
EMBER_AF_RF4CE_MS0_KEY_CODE_LATIN_CAPITAL_LETTER_W
EMBER_AF_RF4CE_MS0_KEY_CODE_LATIN_CAPITAL_LETTER_X
EMBER_AF_RF4CE_MS0_KEY_CODE_LATIN_CAPITAL_LETTER_Y
EMBER_AF_RF4CE_MS0_KEY_CODE_LATIN_CAPITAL_LETTER_Z
EMBER_AF_RF4CE_MS0_KEY_CODE_LATIN_SMALL_LETTER_A
EMBER_AF_RF4CE_MS0_KEY_CODE_LATIN_SMALL_LETTER_B
EMBER_AF_RF4CE_MS0_KEY_CODE_LATIN_SMALL_LETTER_C
EMBER_AF_RF4CE_MS0_KEY_CODE_LATIN_SMALL_LETTER_D
EMBER_AF_RF4CE_MS0_KEY_CODE_LATIN_SMALL_LETTER_E*

EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_SMALL LETTER_F
EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_SMALL LETTER_G
EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_SMALL LETTER_H
EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_SMALL LETTER_I
EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_SMALL LETTER_J
EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_SMALL LETTER_K
EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_SMALL LETTER_L
EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_SMALL LETTER_M
EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_SMALL LETTER_N
EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_SMALL LETTER_O
EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_SMALL LETTER_P
EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_SMALL LETTER_Q
EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_SMALL LETTER_R
EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_SMALL LETTER_S
EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_SMALL LETTER_T
EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_SMALL LETTER_U
EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_SMALL LETTER_V
EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_SMALL LETTER_W
EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_SMALL LETTER_X
EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_SMALL LETTER_Y
EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_SMALL LETTER_Z
EMBER_AF_RF4CE_MSO_KEY_CODE_QUESTION_MARK
EMBER_AF_RF4CE_MSO_KEY_CODE_EXCLAMATION_MARK
EMBER_AF_RF4CE_MSO_KEY_CODE_NUMBER_SIGN
EMBER_AF_RF4CE_MSO_KEY_CODE_DOLLAR_SIGN
EMBER_AF_RF4CE_MSO_KEY_CODE_PERCENT_SIGN
EMBER_AF_RF4CE_MSO_KEY_CODE_AMPERSAND
EMBER_AF_RF4CE_MSO_KEY_CODE_ASTERISK
EMBER_AF_RF4CE_MSO_KEY_CODE_LEFT_PARENTHESIS
EMBER_AF_RF4CE_MSO_KEY_CODE_RIGHT_PARENTHESIS
EMBER_AF_RF4CE_MSO_KEY_CODE_PLUS_SIGN
EMBER_AF_RF4CE_MSO_KEY_CODE_MINUS_SIGN
EMBER_AF_RF4CE_MSO_KEY_CODE_EQUALS_SIGN
EMBER_AF_RF4CE_MSO_KEY_CODE_SLASH
EMBER_AF_RF4CE_MSO_KEY_CODE_UNDERSCORE
EMBER_AF_RF4CE_MSO_KEY_CODE_QUOTATION_MARK
EMBER_AF_RF4CE_MSO_KEY_CODE_COLON
EMBER_AF_RF4CE_MSO_KEY_CODE_SEMICOLON
EMBER_AF_RF4CE_MSO_KEY_CODE_AT_SIGN
EMBER_AF_RF4CE_MSO_KEY_CODE_APOSTROPHE
EMBER_AF_RF4CE_MSO_KEY_CODE_COMMA

Definition at line 63 of file [rf4ce-mso-types.h](#).

8.38.1.5 enum EmberAfRf4ceMsoCheckValidationControl

RF4CE MSO check validation statuses.

Enumerator:

EMBER_AF_RF4CE_MSO_CHECK_VALIDATION_CONTROL_REQUEST_AUTOMATIC_VALIDATION

Definition at line 235 of file [rf4ce-mso-types.h](#).

8.38.1.6 enum EmberAfRf4ceMsoCheckValidationStatus

RF4CE MSO check validation statuses.

Enumerator:

EMBER_AF_RF4CE_MSO_CHECK_VALIDATION_STATUS_SUCCESS The validation is successful.

EMBER_AF_RF4CE_MSO_CHECK_VALIDATION_STATUS_PENDING The validation is still in progress.

EMBER_AF_RF4CE_MSO_CHECK_VALIDATION_STATUS_TIMEOUT The validation timed out, and the binding procedure SHOULD continue with other devices in the list.

EMBER_AF_RF4CE_MSO_CHECK_VALIDATION_STATUS_COLLISION The validation was terminated at the target side, as more than one controller tried to pair.

EMBER_AF_RF4CE_MSO_CHECK_VALIDATION_STATUS_FAILURE The validation failed, and the binding procedure SHOULD continue with other devices in the list.

EMBER_AF_RF4CE_MSO_CHECK_VALIDATION_STATUS_ABORT The validation is aborted, and the binding procedure SHOULD continue with other devices in the list.

EMBER_AF_RF4CE_MSO_CHECK_VALIDATION_STATUS_FULL_ABORT The validation is aborted, and the binding procedure SHOULD NOT continue with other devices in the list.

Definition at line 248 of file [rf4ce-mso-types.h](#).

8.38.1.7 enum EmberAfRf4ceMsoBindingStatus

Enumerator:

EMBER_AF_RF4CE_MSO_BINDING_STATUS_SUCCESS

EMBER_AF_RF4CE_MSO_BINDING_STATUS_NO_VALID_RESPONSE

EMBER_AF_RF4CE_MSO_BINDING_STATUS_NO_VALID_CANDIDATE

EMBER_AF_RF4CE_MSO_BINDING_STATUS_DUPLICATE_CLASS_ABORT

EMBER_AF_RF4CE_MSO_BINDING_STATUS_PAIRING_FAILED

EMBER_AF_RF4CE_MSO_BINDING_STATUS_VALIDATION_TIMEOUT

EMBER_AF_RF4CE_MSO_BINDING_STATUS_VALIDATION_COLLISION

EMBER_AF_RF4CE_MSO_BINDING_STATUS_VALIDATION_FAILURE

EMBER_AF_RF4CE_MSO_BINDING_STATUS_VALIDATION_ABORT

EMBER_AF_RF4CE_MSO_BINDING_STATUS_VALIDATION_FULL_ABORT

Definition at line 279 of file [rf4ce-mso-types.h](#).

8.38.1.8 enum EmberAfRf4ceMsoAttributeId

RF4CE MSO attribute ids.

Enumerator:

EMBER_AF_RF4CE_MSO_ATTRIBUTE_ID_PERIPHERAL_IDS
EMBER_AF_RF4CE_MSO_ATTRIBUTE_ID_RF_STATISTICS
EMBER_AF_RF4CE_MSO_ATTRIBUTE_ID_VERSIONING
EMBER_AF_RF4CE_MSO_ATTRIBUTE_ID_BATTERY_STATUS
EMBER_AF_RF4CE_MSO_ATTRIBUTE_ID_SHORT_RF_RETRY_PERIOD
EMBER_AF_RF4CE_MSO_ATTRIBUTE_ID_IR_RF_DATABASE
EMBER_AF_RF4CE_MSO_ATTRIBUTE_ID_VALIDATION_CONFIGURATION
EMBER_AF_RF4CE_MSO_ATTRIBUTE_ID_GENERAL_PURPOSE

Definition at line 301 of file [rf4ce-mso-types.h](#).

8.38.1.9 enum EmberAfRf4ceMsoIrRfDatabaseFlags

RF4CE MSO IR-RF database entry flags.

Enumerator:

EMBER_AF_RF4CE_MSO_IR_RF_DATABASE_FLAG_NONE No flags.
EMBER_AF_RF4CE_MSO_IR_RF_DATABASE_FLAG_RF_PRESSED_SPECIFIED Indicates that an RF pressed descriptor is included in this attribute, and that an RF message should be generated when this key is pressed. If Use Default is set, this field should be ignored (treated as if it was zero).
EMBER_AF_RF4CE_MSO_IR_RF_DATABASE_FLAG_RF_REPEATED_SPECIFIED Indicates that an RF repeated descriptor is included in this attribute, and that an RF message should be generated when this key is kept pressed. If Use Default is set, this field should be ignored (treated as if it was zero).
EMBER_AF_RF4CE_MSO_IR_RF_DATABASE_FLAG_RF_RELEASED_SPECIFIED Indicates that an RF released descriptor is included in this attribute, and that an RF message should be generated when this key is released. If Use Default is set, this field should be ignored (treated as if it was zero).
EMBER_AF_RF4CE_MSO_IR_RF_DATABASE_FLAG_IR_SPECIFIED Indicates that an IR descriptor is included in this attribute, and that an IR message should be generated when this key is pressed and kept pressed. If Use Default is set, this field should be ignored (treated as if it was zero).
EMBER_AF_RF4CE_MSO_IR_RF_DATABASE_FLAG_DEVICE_TYPE_MASK Represents the device type of the IR descriptor if the IR Specified flag is set.
EMBER_AF_RF4CE_MSO_IR_RF_DATABASE_FLAG_USE_DEFAULT Indicates that the default (known by the RC) RF and IR codes should be used.
EMBER_AF_RF4CE_MSO_IR_RF_DATABASE_FLAG_PERMANENT Indicates that the codes are permanent and can be used for all further presses of this key.

Definition at line 333 of file [rf4ce-mso-types.h](#).

8.38.1.10 enum EmberAfRf4ceMsoIrRfDatabaseDeviceType

RF4CE MSO IR-RF database device types.

Enumerator:

EMBER_AF_RF4CE_MSO_IR_RF_DATABASE_DEVICE_TYPE_TV The device type of the IR descriptor is for a TV.

EMBER_AF_RF4CE_MSO_IR_RF_DATABASE_DEVICE_TYPE_AVR The device type of the IR descriptor is for ab AVR.

Definition at line 383 of file [rf4ce-mso-types.h](#).

8.38.1.11 enum EmberAfRf4ceMsoIrRfDatabaseRfConfig

RF4CE MSO IR-RF database RF config.

Enumerator:

EMBER_AF_RF4CE_MSO_IR_RF_DATABASE_RF_CONFIG_MINIMUM_NUMBER_OF_TRANSMISSIONS_MASK
Indicates the minimum number of transmissions for this code. For acknowledged RF transmissions, this field is set to '1'.

EMBER_AF_RF4CE_MSO_IR_RF_DATABASE_RF_CONFIG_KEEP_TRANSMITTING_UNTIL_KEY_RELEASE
Indicates if the code should continue being transmitted after the minimum number of transmissions have taken place, when the key is kept pressed. (This field only applies to RF Repeated frames.)

EMBER_AF_RF4CE_MSO_IR_RF_DATABASE_RF_CONFIG_SHORT_RETRY Indicates if the RF4CE retry period for UAM messages should be shorted for this code to increase responsiveness of the system.

Definition at line 399 of file [rf4ce-mso-types.h](#).

8.38.1.12 enum EmberAfRf4ceMsoIrRfDatabaseIrConfig

RF4CE MSO IR-RF database IR config.

Enumerator:

EMBER_AF_RF4CE_MSO_IR_RF_DATABASE_IR_CONFIG_MINIMUM_NUMBER_OF_TRANSMISSIONS_MASK
Indicates the minimum number of transmissions for the IR repeat frames. Only valid when Tweak Database is set to '1', otherwise follow behavior as defined by database. Special case: when Tweak Database is set to '1', setting this field to 0xF enforces the use of the value from the database.

EMBER_AF_RF4CE_MSO_IR_RF_DATABASE_IR_CONFIG_KEEP_TRANSMITTING_UNTIL_KEY_RELEASE
Indicates if the repeat frames should continue being transmitted after the minimum number of transmissions as defined by the database have been performed, in case the key remains pressed. Only valid when Tweak Database is set to '1', otherwise follow behavior as defined by database.

EMBER_AF_RF4CE_MSO_IR_RF_DATABASE_IR_CONFIG_SHORT_RETRY Indicates that database behavior should be tweaked, using the "Minimum number of transmissions" and "Keep Transmitting Until Key Release" fields.

Definition at line 434 of file [rf4ce-mso-types.h](#).

8.39 rf4ce-mso-types.h

```

00001 // Copyright 2014 Silicon Laboratories, Inc.
00002
00003 #ifndef __RF4CE_MSO_TYPES_H__
00004 #define __RF4CE_MSO_TYPES_H__
00005
00009 #ifdef DOXYGEN_SHOULD_SKIP_THIS
00010 enum EmberAfRf4ceMsoBindingState
00011 #else
00012 typedef uint8_t EmberAfRf4ceMsoBindingState;
00013 enum
00014 #endif
00015 {
00016     EMBER_AF_RF4CE_MSO_BINDING_STATE_NOT_BOUND
00017         = 0,
00018     EMBER_AF_RF4CE_MSO_BINDING_STATE_BINDING
00019         = 1,
00020     EMBER_AF_RF4CE_MSO_BINDING_STATE_BOUND
00021         = 2,
00022 };
00023
00024 #ifdef DOXYGEN_SHOULD_SKIP_THIS
00025 enum EmberAfRf4ceMsoValidationState
00026 #else
00027 typedef uint8_t EmberAfRf4ceMsoValidationState;
00028 enum
00029 #endif
00030 {
00031     EMBER_AF_RF4CE_MSO_VALIDATION_STATE_NOT_VALIDATED
00032         = 0,
00033     EMBER_AF_RF4CE_MSO_VALIDATION_STATE_REJECTED
00034         = 1,
00035     EMBER_AF_RF4CE_MSO_VALIDATION_STATE_VALIDATING
00036         = 2,
00037     EMBER_AF_RF4CE_MSO_VALIDATION_STATE_REVALIDATING
00038         = 3,
00039     EMBER_AF_RF4CE_MSO_VALIDATION_STATE_VALIDATED
00040         = 4,
00041 };
00042
00043 #ifdef DOXYGEN_SHOULD_SKIP_THIS
00044 enum EmberAfRf4ceMsoCommandCode
00045 #else
00046 typedef uint8_t EmberAfRf4ceMsoCommandCode;
00047 enum
00048 {
00049     EMBER_AF_RF4CE_MSO_COMMAND_USER_CONTROL_PRESSED
00050         = 0x01,
00051     EMBER_AF_RF4CE_MSO_COMMAND_USER_CONTROL_REPEAT
00052         = 0x02,
00053     EMBER_AF_RF4CE_MSO_COMMAND_USER_CONTROL_RELEASED
00054         = 0x03,
00055     EMBER_AF_RF4CE_MSO_COMMAND_CHECK_VALIDATION_REQUEST
00056         = 0x20,
00057     EMBER_AF_RF4CE_MSO_COMMAND_CHECK_VALIDATION_RESPONSE
00058         = 0x21,
00059     EMBER_AF_RF4CE_MSO_COMMAND_SET_ATTRIBUTE_REQUEST
00060         = 0x22,
00061     EMBER_AF_RF4CE_MSO_COMMAND_SET_ATTRIBUTE_RESPONSE
00062         = 0x23,
00063     EMBER_AF_RF4CE_MSO_COMMAND_GET_ATTRIBUTE_REQUEST
00064         = 0x24,
00065     EMBER_AF_RF4CE_MSO_COMMAND_GET_ATTRIBUTE_RESPONSE
00066         = 0x25,
00067 };
00068
00069 #ifdef DOXYGEN_SHOULD_SKIP_THIS
00070 enum EmberAfRf4ceMsoKeyCode
00071 #else
00072 typedef uint8_t EmberAfRf4ceMsoKeyCode;
00073 enum
00074 #endif
00075 {
00076     EMBER_AF_RF4CE_MSO_KEY_CODE_OK
00077         = 0x00,
00078     EMBER_AF_RF4CE_MSO_KEY_CODE_UP_ARROW
00079         = 0x01,

```

```

00071 EMBER_AF_RF4CE_MSO_KEY_CODE_DOWN_ARROW
                  = 0x02,
00072 EMBER_AF_RF4CE_MSO_KEY_CODE_LEFT_ARROW
                  = 0x03,
00073 EMBER_AF_RF4CE_MSO_KEY_CODE_RIGHT_ARROW
                  = 0x04,
00074 EMBER_AF_RF4CE_MSO_KEY_CODE_MENU
                  = 0x09,
00075 EMBER_AF_RF4CE_MSO_KEY_CODE_DVR
                  = 0x0B,
00076 EMBER_AF_RF4CE_MSO_KEY_CODE_FAV
                  = 0x0C,
00077 EMBER_AF_RF4CE_MSO_KEY_CODE_EXIT
                  = 0x0D,
00078 EMBER_AF_RF4CE_MSO_KEY_CODE_HOME
                  = 0x10,
00079 EMBER_AF_RF4CE_MSO_KEY_CODE_DIGIT_0
                  = 0x20,
00080 EMBER_AF_RF4CE_MSO_KEY_CODE_DIGIT_1
                  = 0x21,
00081 EMBER_AF_RF4CE_MSO_KEY_CODE_DIGIT_2
                  = 0x22,
00082 EMBER_AF_RF4CE_MSO_KEY_CODE_DIGIT_3
                  = 0x23,
00083 EMBER_AF_RF4CE_MSO_KEY_CODE_DIGIT_4
                  = 0x24,
00084 EMBER_AF_RF4CE_MSO_KEY_CODE_DIGIT_5
                  = 0x25,
00085 EMBER_AF_RF4CE_MSO_KEY_CODE_DIGIT_6
                  = 0x26,
00086 EMBER_AF_RF4CE_MSO_KEY_CODE_DIGIT_7
                  = 0x27,
00087 EMBER_AF_RF4CE_MSO_KEY_CODE_DIGIT_8
                  = 0x28,
00088 EMBER_AF_RF4CE_MSO_KEY_CODE_DIGIT_9
                  = 0x29,
00089 EMBER_AF_RF4CE_MSO_KEY_CODE_FULL_STOP
                  = 0x2A,
00090 EMBER_AF_RF4CE_MSO_KEY_CODE_RETURN
                  = 0x2B,
00091 EMBER_AF_RF4CE_MSO_KEY_CODE_CHANNEL_UP
                  = 0x30,
00092 EMBER_AF_RF4CE_MSO_KEY_CODE_CHANNEL_DOWN
                  = 0x31,
00093 EMBER_AF_RF4CE_MSO_KEY_CODE_LAST
                  = 0x32,
00094 EMBER_AF_RF4CE_MSO_KEY_CODE_LANG
                  = 0x33,
00095 EMBER_AF_RF4CE_MSO_KEY_CODE_INPUT_SELECT
                  = 0x34,
00096 EMBER_AF_RF4CE_MSO_KEY_CODE_INFO
                  = 0x35,
00097 EMBER_AF_RF4CE_MSO_KEY_CODE_HELP
                  = 0x36,
00098 EMBER_AF_RF4CE_MSO_KEY_CODE_PAGE_UP
                  = 0x37,
00099 EMBER_AF_RF4CE_MSO_KEY_CODE_PAGE_DOWN
                  = 0x38,
00100 EMBER_AF_RF4CE_MSO_KEY_CODE_MOTION
                  = 0x3B,
00101 EMBER_AF_RF4CE_MSO_KEY_CODE_SEARCH
                  = 0x3C,
00102 EMBER_AF_RF4CE_MSO_KEY_CODE_LIVE
                  = 0x3D,
00103 EMBER_AF_RF4CE_MSO_KEY_CODE_HD_ZOOM
                  = 0x3E,
00104 EMBER_AF_RF4CE_MSO_KEY_CODE_SHARE
                  = 0x3F,
00105 EMBER_AF_RF4CE_MSO_KEY_CODE_TV_POWER
                  = 0x40,
00106 EMBER_AF_RF4CE_MSO_KEY_CODE_VOLUME_UP
                  = 0x41,
00107 EMBER_AF_RF4CE_MSO_KEY_CODE_VOLUME_DOWN
                  = 0x42,
00108 EMBER_AF_RF4CE_MSO_KEY_CODE_MUTE
                  = 0x43,
00109 EMBER_AF_RF4CE_MSO_KEY_CODE_PLAY
                  = 0x44,
00110 EMBER_AF_RF4CE_MSO_KEY_CODE_STOP
                  = 0x45,

```

```

00111 EMBER_AF_RF4CE_MSO_KEY_CODE_PAUSE
00112             = 0x46,
00113 EMBER_AF_RF4CE_MSO_KEY_CODE_RECORD
00114             = 0x47,
00115 EMBER_AF_RF4CE_MSO_KEY_CODE_REWIND
00116             = 0x48,
00117 EMBER_AF_RF4CE_MSO_KEY_CODE_FAST_FORWARD
00118             = 0x49,
00119 EMBER_AF_RF4CE_MSO_KEY_CODE_30_SECOND_SKIP_AHEAD
00120             = 0x4B,
00121 EMBER_AF_RF4CE_MSO_KEY_CODE_REPLY
00122             = 0x4C,
00123 EMBER_AF_RF4CE_MSO_KEY_CODE_SWAP
00124             = 0x51,
00125 EMBER_AF_RF4CE_MSO_KEY_CODE_ON_DEMAND
00126             = 0x52,
00127 EMBER_AF_RF4CE_MSO_KEY_CODE_GUIDE
00128             = 0x53,
00129 EMBER_AF_RF4CE_MSO_KEY_CODE_PUSH_TO_TALK
00130             = 0x57,
00131 EMBER_AF_RF4CE_MSO_KEY_CODE_PIP_ON_OFF
00132             = 0x58,
00133 EMBER_AF_RF4CE_MSO_KEY_CODE_PIP_MOVE
00134             = 0x59,
00135 EMBER_AF_RF4CE_MSO_KEY_CODE_PIP_CHANNEL_UP
00136             = 0x5A,
00137 EMBER_AF_RF4CE_MSO_KEY_CODE_PIP_CHANNEL_DOWN
00138             = 0x5B,
00139 EMBER_AF_RF4CE_MSO_KEY_CODE_LOCK
00140             = 0x5C,
00141 EMBER_AF_RF4CE_MSO_KEY_CODE_DAY_UP
00142             = 0x5D,
00143 EMBER_AF_RF4CE_MSO_KEY_CODE_DAY_DOWN
00144             = 0x5E,
00145 EMBER_AF_RF4CE_MSO_KEY_CODE_PLAY_PAUSE
00146             = 0x61,
00147 EMBER_AF_RF4CE_MSO_KEY_CODE_STOP_VIDEO
00148             = 0x64,
00149 EMBER_AF_RF4CE_MSO_KEY_CODE_MUTE_MIC
00150             = 0x65,
00151 EMBER_AF_RF4CE_MSO_KEY_CODE_POWER_TOGGLE
00152             = 0x66,
00153 EMBER_AF_RF4CE_MSO_KEY_CODE_POWER_OFF
00154             = 0x67,
00155 EMBER_AF_RF4CE_MSO_KEY_CODE_POWER_ON
00156             = 0x6D,
00157 EMBER_AF_RF4CE_MSO_KEY_CODE_OCAP_B
00158             = 0x71,
00159 EMBER_AF_RF4CE_MSO_KEY_CODE_BLUE_SQUARE
00160             = 0x71,
00161 EMBER_AF_RF4CE_MSO_KEY_CODE_OCAP_C
00162             = 0x72,
00163 EMBER_AF_RF4CE_MSO_KEY_CODE_RED_CIRCLE
00164             = 0x72,
00165 EMBER_AF_RF4CE_MSO_KEY_CODE_OCAP_D
00166             = 0x73,
00167 EMBER_AF_RF4CE_MSO_KEY_CODE_GREEN_DIAMOND
00168             = 0x73,
00169 EMBER_AF_RF4CE_MSO_KEY_CODE_OCAP_A
00170             = 0x74,
00171 EMBER_AF_RF4CE_MSO_KEY_CODE_YELLOW_TRIANGLE
00172             = 0x74,
00173 EMBER_AF_RF4CE_MSO_KEY_CODE_PROFILE
00174             = 0xA0,
00175 EMBER_AF_RF4CE_MSO_KEY_CODE_CALL
00176             = 0xA1,
00177 EMBER_AF_RF4CE_MSO_KEY_CODE_HOLD
00178             = 0xA2,
00179 EMBER_AF_RF4CE_MSO_KEY_CODE_END
00180             = 0xA3,
00181 EMBER_AF_RF4CE_MSO_KEY_CODE_VIEWS
00182             = 0xA4,
00183 EMBER_AF_RF4CE_MSO_KEY_CODE_SELF_VIEW
00184             = 0xA5,
00185 EMBER_AF_RF4CE_MSO_KEY_CODE_ZOOM_IN
00186             = 0xA6,
00187 EMBER_AF_RF4CE_MSO_KEY_CODE_ZOOM_OUT
00188             = 0xA7,
00189 EMBER_AF_RF4CE_MSO_KEY_CODE_BACKSPACE
00190             = 0xA8,

```

```

00151 EMBER_AF_RF4CE_MSO_KEY_CODE_LOCK_UNLOCK
          = 0xA9,
00152 EMBER_AF_RF4CE_MSO_KEY_CODE_CAPS
          = 0xAA,
00153 EMBER_AF_RF4CE_MSO_KEY_CODE_ALT
          = 0xAB,
00154 EMBER_AF_RF4CE_MSO_KEY_CODE_SPACE
          = 0xAC,
00155 EMBER_AF_RF4CE_MSO_KEY_CODE_WWW_DOT
          = 0xAD,
00156 EMBER_AF_RF4CE_MSO_KEY_CODE_DOT_COM
          = 0xAE,
00157 EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_CAPITAL_LETTER_A
          = 0xB0,
00158 EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_CAPITAL_LETTER_B
          = 0xB1,
00159 EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_CAPITAL_LETTER_C
          = 0xB2,
00160 EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_CAPITAL_LETTER_D
          = 0xB3,
00161 EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_CAPITAL_LETTER_E
          = 0xB4,
00162 EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_CAPITAL_LETTER_F
          = 0xB5,
00163 EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_CAPITAL_LETTER_G
          = 0xB6,
00164 EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_CAPITAL_LETTER_H
          = 0xB7,
00165 EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_CAPITAL_LETTER_I
          = 0xB8,
00166 EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_CAPITAL_LETTER_J
          = 0xB9,
00167 EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_CAPITAL_LETTER_K
          = 0xBA,
00168 EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_CAPITAL_LETTER_L
          = 0xBB,
00169 EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_CAPITAL_LETTER_M
          = 0xBC,
00170 EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_CAPITAL_LETTER_N
          = 0xBD,
00171 EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_CAPITAL_LETTER_O
          = 0xBE,
00172 EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_CAPITAL_LETTER_P
          = 0xBF,
00173 EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_CAPITAL_LETTER_Q
          = 0xC0,
00174 EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_CAPITAL_LETTER_R
          = 0xC1,
00175 EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_CAPITAL_LETTER_S
          = 0xC2,
00176 EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_CAPITAL_LETTER_T
          = 0xC3,
00177 EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_CAPITAL_LETTER_U
          = 0xC4,
00178 EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_CAPITAL_LETTER_V
          = 0xC5,
00179 EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_CAPITAL_LETTER_W
          = 0xC6,
00180 EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_CAPITAL_LETTER_X
          = 0xC7,
00181 EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_CAPITAL_LETTER_Y
          = 0xC8,
00182 EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_CAPITAL_LETTER_Z
          = 0xC9,
00183 EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_SMALL_LETTER_A
          = 0xCA,
00184 EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_SMALL_LETTER_B
          = 0xCB,
00185 EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_SMALL_LETTER_C
          = 0xCC,
00186 EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_SMALL_LETTER_D
          = 0xCD,
00187 EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_SMALL_LETTER_E
          = 0xCE,
00188 EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_SMALL_LETTER_F
          = 0xCF,
00189 EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_SMALL_LETTER_G
          = 0xD0,
00190 EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_SMALL_LETTER_H
          = 0xD1,

```

```

00191 EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_SMALL LETTER_I
00192 = 0xD2,
00193 EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_SMALL LETTER_J
00194 = 0xD3,
00195 EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_SMALL LETTER_K
00196 = 0xD4,
00197 EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_SMALL LETTER_L
00198 = 0xD5,
00199 EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_SMALL LETTER_M
00200 = 0xD6,
00201 EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_SMALL LETTER_N
00202 = 0xD7,
00203 EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_SMALL LETTER_O
00204 = 0xD8,
00205 EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_SMALL LETTER_P
00206 = 0xD9,
00207 EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_SMALL LETTER_Q
00208 = 0xDA,
00209 EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_SMALL LETTER_R
00210 = 0xDB,
00211 EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_SMALL LETTER_S
00212 = 0xDC,
00213 EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_SMALL LETTER_T
00214 = 0xDD,
00215 EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_SMALL LETTER_U
00216 = 0xDE,
00217 EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_SMALL LETTER_V
00218 = 0xDF,
00219 EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_SMALL LETTER_W
00220 = 0xE0,
00221 EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_SMALL LETTER_X
00222 = 0xE1,
00223 EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_SMALL LETTER_Y
00224 = 0xE2,
00225 EMBER_AF_RF4CE_MSO_KEY_CODE_LATIN_SMALL LETTER_Z
00226 = 0xE3,
00227 EMBER_AF_RF4CE_MSO_KEY_CODE_QUESTION_MARK
00228 = 0xE4,
00229 EMBER_AF_RF4CE_MSO_KEY_CODE_EXCLAMATION_MARK
00230 = 0xE5,
00231 EMBER_AF_RF4CE_MSO_KEY_CODE_NUMBER_SIGN
00232 = 0xE6,
00233 EMBER_AF_RF4CE_MSO_KEY_CODE_DOLLAR_SIGN
00234 = 0xE7,
00235 EMBER_AF_RF4CE_MSO_KEY_CODE_PERCENT_SIGN
00236 = 0xE8,
00237 EMBER_AF_RF4CE_MSO_KEY_CODE_AMPERSAND
00238 = 0xE9,
00239 EMBER_AF_RF4CE_MSO_KEY_CODE_ASTERISK
00240 = 0xEA,
00241 EMBER_AF_RF4CE_MSO_KEY_CODE_LEFT_PARENTHESIS
00242 = 0xEB,
00243 EMBER_AF_RF4CE_MSO_KEY_CODE_RIGHT_PARENTHESIS
00244 = 0xEC,
00245 EMBER_AF_RF4CE_MSO_KEY_CODE_PLUS_SIGN
00246 = 0xED,
00247 EMBER_AF_RF4CE_MSO_KEY_CODE_MINUS_SIGN
00248 = 0xEE,
00249 EMBER_AF_RF4CE_MSO_KEY_CODE_EQUALS_SIGN
00250 = 0xEF,
00251 EMBER_AF_RF4CE_MSO_KEY_CODE_SLASH
00252 = 0xF0,
00253 EMBER_AF_RF4CE_MSO_KEY_CODE_UNDERSCORE
00254 = 0xF1,
00255 EMBER_AF_RF4CE_MSO_KEY_CODE_QUOTATION_MARK
00256 = 0xF2,
00257 EMBER_AF_RF4CE_MSO_KEY_CODE_COLON
00258 = 0xF3,
00259 EMBER_AF_RF4CE_MSO_KEY_CODE_SEMICOLON
00260 = 0xF4,
00261 EMBER_AF_RF4CE_MSO_KEY_CODE_AT_SIGN
00262 = 0xF5,
00263 EMBER_AF_RF4CE_MSO_KEY_CODE_APOSTROPHE
00264 = 0xF6,
00265 EMBER_AF_RF4CE_MSO_KEY_CODE_COMMA
00266 = 0xF7,
00267 };
00268 #ifdef DOXYGEN_SHOULD_SKIP_THIS
00269 enum EmberAfRf4ceMsoCheckValidationControl

```

```

00236 #else
00237 typedef uint8_t EmberAfRf4ceMsoCheckValidationControl
00238 ;
00239 enum
00240 #endif
00241 {
00241   EMBER_AF_RF4CE_MSO_CHECK_VALIDATION_CONTROL_REQUEST_AUTOMATIC_VALIDATION
00241   = BIT(0)
00242 };
00243
00247 #ifdef DOXYGEN_SHOULD_SKIP_THIS
00248 enum EmberAfRf4ceMsoCheckValidationStatus
00249 #else
00250 typedef uint8_t EmberAfRf4ceMsoCheckValidationStatus
00251 ;
00251 enum
00252 #endif
00253 {
00255   EMBER_AF_RF4CE_MSO_CHECK_VALIDATION_STATUS_SUCCESS
00255   = 0x00,
00257   EMBER_AF_RF4CE_MSO_CHECK_VALIDATION_STATUS_PENDING
00257   = 0xC0,
00260   EMBER_AF_RF4CE_MSO_CHECK_VALIDATION_STATUS_TIMEOUT
00260   = 0xC1,
00263   EMBER_AF_RF4CE_MSO_CHECK_VALIDATION_STATUS_COLLISION
00263   = 0xC2,
00266   EMBER_AF_RF4CE_MSO_CHECK_VALIDATION_STATUS_FAILURE
00266   = 0xC3,
00269   EMBER_AF_RF4CE_MSO_CHECK_VALIDATION_STATUS_ABORT
00269   = 0xC4,
00272   EMBER_AF_RF4CE_MSO_CHECK_VALIDATION_STATUS_FULL_ABORT
00272   = 0xC5,
00273 };
00274
00275 /*
00276 * @brief RF4CE MSO binding statuses.
00277 */
00278 #ifdef DOXYGEN_SHOULD_SKIP_THIS
00279 enum EmberAfRf4ceMsoBindingStatus
00280 #else
00281 typedef uint8_t EmberAfRf4ceMsoBindingStatus;
00282 enum
00283 #endif
00284 {
00285   EMBER_AF_RF4CE_MSO_BINDING_STATUS_SUCCESS
00285   = 0x00,
00286   EMBER_AF_RF4CE_MSO_BINDING_STATUS_NO_VALID_RESPONSE
00286   = 0x01,
00287   EMBER_AF_RF4CE_MSO_BINDING_STATUS_NO_VALID_CANDIDATE
00287   = 0x02,
00288   EMBER_AF_RF4CE_MSO_BINDING_STATUS_DUPLICATE_CLASS_ABORT
00288   = 0x03,
00289   EMBER_AF_RF4CE_MSO_BINDING_STATUS_PAIRING_FAILED
00289   = 0x04,
00290   EMBER_AF_RF4CE_MSO_BINDING_STATUS_VALIDATION_TIMEOUT
00290   = EMBER_AF_RF4CE_MSO_CHECK_VALIDATION_STATUS_TIMEOUT
00291   EMBER_AF_RF4CE_MSO_BINDING_STATUS_VALIDATION_COLLISION
00291   = EMBER_AF_RF4CE_MSO_CHECK_VALIDATION_STATUS_COLLISION
00292   EMBER_AF_RF4CE_MSO_BINDING_STATUS_VALIDATION_FAILURE
00292   = EMBER_AF_RF4CE_MSO_CHECK_VALIDATION_STATUS_FAILURE
00293   EMBER_AF_RF4CE_MSO_BINDING_STATUS_VALIDATION_ABORT
00293   = EMBER_AF_RF4CE_MSO_CHECK_VALIDATION_STATUS_ABORT
00294   EMBER_AF_RF4CE_MSO_BINDING_STATUS_VALIDATION_FULL_ABORT
00294   = EMBER_AF_RF4CE_MSO_CHECK_VALIDATION_STATUS_FULL_ABORT
00295 };
00296
00300 #ifdef DOXYGEN_SHOULD_SKIP_THIS
00301 enum EmberAfRf4ceMsoAttributeId
00302 #else
00303 typedef uint8_t EmberAfRf4ceMsoAttributeId;
00304 enum
00305 #endif
00306 {
00307   EMBER_AF_RF4CE_MSO_ATTRIBUTE_ID_PERIPHERAL_IDS
00307   = 0x00,

```

```

00308     EMBER_AF_RF4CE_MSO_ATTRIBUTE_ID_RF_STATISTICS
00309         = 0x01,
00310     EMBER_AF_RF4CE_MSO_ATTRIBUTE_ID_VERSIONING
00311         = 0x02,
00312     EMBER_AF_RF4CE_MSO_ATTRIBUTE_ID_BATTERY_STATUS
00313         = 0x03,
00314     EMBER_AF_RF4CE_MSO_ATTRIBUTE_ID_SHORT_RF_RETRY_PERIOD
00315         = 0x04,
00316     EMBER_AF_RF4CE_MSO_ATTRIBUTE_ID_IR_RF_DATABASE
00317         = 0xDB,
00318     EMBER_AF_RF4CE_MSO_ATTRIBUTE_ID_VALIDATION_CONFIGURATION
00319         = 0xDC,
00320     EMBER_AF_RF4CE_MSO_ATTRIBUTE_ID_GENERAL_PURPOSE
00321         = 0xFF,
00322 };
00323
00324 typedef struct {
00325     uint8_t pairingIndex;
00326     EmberAfRf4ceMsoCommandCode commandCode;
00327     EmberAfRf4ceMsoKeyCode rcCommandCode;
00328     const uint8_t *rcCommandPayload;
00329     uint8_t rcCommandPayloadLength;
00330     uint16_t timeMs;
00331 } EmberAfRf4ceMsoUserControlRecord;
00332
00333 #ifdef DOXYGEN_SHOULD_SKIP_THIS
00334 enum EmberAfRf4ceMsoIrRfDatabaseFlags
00335 #else
00336 typedef uint8_t EmberAfRf4ceMsoIrRfDatabaseFlags
00337 ;
00338
00339 {
00340     EMBER_AF_RF4CE_MSO_IR_RF_DATABASE_FLAG_NONE
00341         = 0x00,
00342     EMBER_AF_RF4CE_MSO_IR_RF_DATABASE_FLAG_RF_PRESSED_SPECIFIED
00343         = 0x01,
00344     EMBER_AF_RF4CE_MSO_IR_RF_DATABASE_FLAG_RF_REPEAT_SPECIFIED
00345         = 0x02,
00346     EMBER_AF_RF4CE_MSO_IR_RF_DATABASE_FLAG_RF_RELEASED_SPECIFIED
00347         = 0x04,
00348     EMBER_AF_RF4CE_MSO_IR_RF_DATABASE_FLAG_IR_SPECIFIED
00349         = 0x08,
00350     EMBER_AF_RF4CE_MSO_IR_RF_DATABASE_FLAG_DEVICE_TYPE_MASK
00351         = 0x48,
00352     EMBER_AF_RF4CE_MSO_IR_RF_DATABASE_FLAG_USE_DEFAULT
00353         = 0x40,
00354     EMBER_AF_RF4CE_MSO_IR_RF_DATABASE_FLAG_PERMANENT
00355         = 0x80,
00356 };
00357
00358 #ifdef DOXYGEN_SHOULD_SKIP_THIS
00359 enum EmberAfRf4ceMsoIrRfDatabaseDeviceType
00360 #else
00361 typedef uint8_t EmberAfRf4ceMsoIrRfDatabaseDeviceType
00362 ;
00363
00364 enum
00365 #endif
00366 {
00367     EMBER_AF_RF4CE_MSO_IR_RF_DATABASE_DEVICE_TYPE_TV
00368         = 0x00,
00369     EMBER_AF_RF4CE_MSO_IR_RF_DATABASE_DEVICE_TYPE_AVR
00370         = 0x20,
00371 };
00372
00373 #ifdef DOXYGEN_SHOULD_SKIP_THIS
00374 enum EmberAfRf4ceMsoIrRfDatabaseRfConfig
00375 #else
00376 typedef uint8_t EmberAfRf4ceMsoIrRfDatabaseRfConfig
00377 ;
00378
00379 enum
00380 #endif
00381 {
00382     EMBER_AF_RF4CE_MSO_IR_RF_DATABASE_RF_CONFIG_MINIMUM_NUMBER_OF_TRANSMISSIONS_MASK
00383         = 0x0F,
00384     EMBER_AF_RF4CE_MSO_IR_RF_DATABASE_RF_CONFIG_KEEP_TRANSMITTING_UNTIL_KEY_RELEASE
00385         = 0x10,

```

```

00417     EMBER_AF_RF4CE_MSO_IR_RF_DATABASE_RF_CONFIG_SHORT_RETRY
00418     = 0x20,
00419
00420     typedef struct {
00421         EmberAfRf4ceMsoIrRfDatabaseRfConfig
00422             rfConfig;
00423         EmberRf4ceTxOption txOptions;
00424         uint8_t payloadLength;
00425         const uint8_t *payload;
00426     } EmberAfRf4ceMsoIrRfDatabaseRfDescriptor
00427 ;
00428
00429 #ifdef DOXYGEN_SHOULD_SKIP_THIS
00430     enum EmberAfRf4ceMsoIrRfDatabaseIrConfig
00431 #else
00432     typedef uint8_t EmberAfRf4ceMsoIrRfDatabaseIrConfig
00433 ;
00434 enum
00435 #endif
00436 {
00437
00438     EMBER_AF_RF4CE_MSO_IR_RF_DATABASE_IR_CONFIG_MINIMUM_NUMBER_OF_TRANSMISSIONS_MASK
00439     = 0x0F,
00440
00441     EMBER_AF_RF4CE_MSO_IR_RF_DATABASE_IR_CONFIG_KEEP_TRANSMITTING_UNTIL_KEY_RELEASE
00442     = 0x10,
00443     EMBER_AF_RF4CE_MSO_IR_RF_DATABASE_IR_CONFIG_SHORT_RETRY
00444     = 0x40,
00445 };
00446
00447     typedef struct {
00448         EmberAfRf4ceMsoIrRfDatabaseIrConfig
00449             irConfig;
00450         uint8_t irCodeLength;
00451         const uint8_t *irCode;
00452     } EmberAfRf4ceMsoIrRfDatabaseIrDescriptor
00453 ;
00454
00455     typedef struct {
00456         EmberAfRf4ceMsoIrRfDatabaseFlags flags;
00457         EmberAfRf4ceMsoIrRfDatabaseRfDescriptor
00458             rfPressedDescriptor;
00459         EmberAfRf4ceMsoIrRfDatabaseRfDescriptor
00460             rfRepeatedDescriptor;
00461         EmberAfRf4ceMsoIrRfDatabaseRfDescriptor
00462             rfReleasedDescriptor;
00463         EmberAfRf4ceMsoIrRfDatabaseIrDescriptor
00464             irDescriptor;
00465     } EmberAfRf4ceMsoIrRfDatabaseEntry;
00466
00467 #endif // __RF4CE_MSO_TYPES_H__

```

8.40 rf4ce-mso.h File Reference

```
#include "rf4ce-mso-types.h"
```

Macros

- #define EMBER_AF_PLUGIN_RF4CE_MSO_IS_RECIPIENT

Functions

- EmberStatus emberAfRf4ceMsoBind (void)
- EmberStatus emberAfRf4ceMsoWatchdogKick (uint16_t validationWatchdogTimeMs)
- EmberStatus emberAfRf4ceMsoValidate (void)

- EmberStatus `emberAfRf4ceMsoTerminateValidation` (void)
- EmberStatus `emberAfRf4ceMsoAbortValidation` (bool fullAbort)
- EmberStatus `emberAfRf4ceMsoUserControlPress` (uint8_t pairingIndex, `EmberAfRf4ceMsoKeyCode` rcCommandCode, const uint8_t *rcCommandPayload, uint8_t rcCommandPayloadLength, bool atomic)
- EmberStatus `emberAfRf4ceMsoUserControlRelease` (uint8_t pairingIndex, `EmberAfRf4ceMsoKeyCode` rcCommandCode)
- EmberStatus `emberAfRf4ceMsoSetAttributeRequest` (uint8_t pairingIndex, `EmberAfRf4ceMsoAttributeId` attributeId, uint8_t index, uint8_t valueLen, const uint8_t *value)
- EmberStatus `emberAfRf4ceMsoGetAttributeRequest` (uint8_t pairingIndex, `EmberAfRf4ceMsoAttributeId` attributeId, uint8_t index, uint8_t valueLen)
- bool `emberAfRf4ceMsoIrRfDatabaseEntryUseDefault` (const `EmberAfRf4ceMsoIrRfDatabaseEntry` *entry)
- bool `emberAfRf4ceMsoIrRfDatabaseEntryHasRfPressedDescriptor` (const `EmberAfRf4ceMsoIrRfDatabaseEntry` *entry)
- bool `emberAfRf4ceMsoIrRfDatabaseEntryHasRfRepeatedDescriptor` (const `EmberAfRf4ceMsoIrRfDatabaseEntry` *entry)
- bool `emberAfRf4ceMsoIrRfDatabaseEntryHasRfReleasedDescriptor` (const `EmberAfRf4ceMsoIrRfDatabaseEntry` *entry)
- uint8_t `emberAfRf4ceMsoIrRfDatabaseEntryGetMinimumNumberOfTransmissions` (const `EmberAfRf4ceMsoIrRfDatabaseRfDescriptor` *rfDescriptor)
- bool `emberAfRf4ceMsoIrRfDatabaseEntryShouldTransmitUntilRelease` (const `EmberAfRf4ceMsoIrRfDatabaseRfDescriptor` *rfDescriptor)
- bool `emberAfRf4ceMsoIrRfDatabaseEntryHasIrDescriptor` (const `EmberAfRf4ceMsoIrRfDatabaseEntry` *entry)

8.41 rf4ce-mso.h

```

00001 // Copyright 2014 Silicon Laboratories, Inc.
00002
00003 #ifndef __RF4CE_MSO_H__
00004 #define __RF4CE_MSO_H__
00005
00006 #include "rf4ce-mso-types.h"
00007
00008 // Only controllers can be originators in MSO and only targets can be
00009 // recipients.
00010 #ifdef EMBER_AF_RF4CE_NODE_TYPE_CONTROLLER
00011     #define EMBER_AF_PLUGIN_RF4CE_MSO_IS_ORIGINATOR
00012 #else
00013     #define EMBER_AF_PLUGIN_RF4CE_MSO_IS_RECIPIENT
00014 #endif
00015
00016 EmberStatus emberAfRf4ceMsoBind(void);
00017
00018 EmberStatus emberAfRf4ceMsoWatchdogKick(uint16_t
00019     validationWatchdogTimeMs);
00020
00021 EmberStatus emberAfRf4ceMsoValidate(void);
00022
00023 EmberStatus emberAfRf4ceMsoTerminateValidation
00024     (void);
00025
00026 EmberStatus emberAfRf4ceMsoAbortValidation(bool
00027     fullAbort);
00028
00029 EmberStatus emberAfRf4ceMsoUserControlPress(
00030     uint8_t pairingIndex,
00031             EmberAfRf4ceMsoKeyCode
00032     rcCommandCode,
00033             const uint8_t *rcCommandPayload,
00034

```

```

00218                     uint8_t rcCommandPayloadLength,
00219                     bool atomic);
00220
00235 EmberStatus emberAfRf4ceMsoUserControlRelease(
00236     uint8_t pairingIndex,
00237             EmberAfRf4ceMsoKeyCode
00238     rcCommandCode);
00237
00258 EmberStatus emberAfRf4ceMsoSetAttributeRequest
00259     (uint8_t pairingIndex,
00260         EmberAfRf4ceMsoAttributeId
00261         attributeId,
00262             uint8_t index,
00263             uint8_t valueLen,
00264             const uint8_t *value);
00263
00283 EmberStatus emberAfRf4ceMsoGetAttributeRequest
00284     (uint8_t pairingIndex,
00285         EmberAfRf4ceMsoAttributeId
00286         attributeId,
00287             uint8_t index,
00288             uint8_t valueLen);
00287
00288 #ifdef DOXYGEN_SHOULD_SKIP_THIS
00289
00296     bool emberAfRf4ceMsoIrRfDatabaseEntryUseDefault
00297     (const EmberAfRf4ceMsoIrRfDatabaseEntry *entry);
00297 #else
00298     #define emberAfRf4ceMsoIrRfDatabaseEntryUseDefault(entry) \
00299         (READBITS((entry)->flags, \
00300             EMBER_AF_RF4CE_MSO_IR_RF_DATABASE_FLAG_USE_DEFAULT) \
00301             == EMBER_AF_RF4CE_MSO_IR_RF_DATABASE_FLAG_USE_DEFAULT)
00302 #endif
00303
00304 #ifdef DOXYGEN_SHOULD_SKIP_THIS
00305
00313     bool emberAfRf4ceMsoIrRfDatabaseEntryHasRfPressedDescriptor
00314     (const EmberAfRf4ceMsoIrRfDatabaseEntry *entry);
00314 #else
00315     #define emberAfRf4ceMsoIrRfDatabaseEntryHasRfPressedDescriptor(entry) \
00316         (READBITS((entry)->flags, \
00317             (EMBER_AF_RF4CE_MSO_IR_RF_DATABASE_FLAG_RF_PRESSED_SPECIFIED \
00318                 | EMBER_AF_RF4CE_MSO_IR_RF_DATABASE_FLAG_USE_DEFAULT) \
00319                 == EMBER_AF_RF4CE_MSO_IR_RF_DATABASE_FLAG_RF_PRESSED_SPECIFIED)
00320 #endif
00321
00322 #ifdef DOXYGEN_SHOULD_SKIP_THIS
00323
00331     bool emberAfRf4ceMsoIrRfDatabaseEntryHasRfRepeatedDescriptor
00332     (const EmberAfRf4ceMsoIrRfDatabaseEntry *entry);
00332 #else
00333     #define emberAfRf4ceMsoIrRfDatabaseEntryHasRfRepeatedDescriptor(entry) \
00334         (READBITS((entry)->flags, \
00335             (EMBER_AF_RF4CE_MSO_IR_RF_DATABASE_FLAG_RF_REPEAT_SPECIFIED \
00336                 | EMBER_AF_RF4CE_MSO_IR_RF_DATABASE_FLAG_USE_DEFAULT) \
00337                 == EMBER_AF_RF4CE_MSO_IR_RF_DATABASE_FLAG_RF_REPEAT_SPECIFIED)
00338 #endif
00339
00340 #ifdef DOXYGEN_SHOULD_SKIP_THIS
00341
00349     bool emberAfRf4ceMsoIrRfDatabaseEntryHasRfReleasedDescriptor
00350     (const EmberAfRf4ceMsoIrRfDatabaseEntry *entry);
00350 #else
00351     #define emberAfRf4ceMsoIrRfDatabaseEntryHasRfReleasedDescriptor(entry) \
00352         (READBITS((entry)->flags, \
00353             (EMBER_AF_RF4CE_MSO_IR_RF_DATABASE_FLAG_RF_RELEASED_SPECIFIED \
00354                 | EMBER_AF_RF4CE_MSO_IR_RF_DATABASE_FLAG_USE_DEFAULT) \
00355                 == EMBER_AF_RF4CE_MSO_IR_RF_DATABASE_FLAG_RF_RELEASED_SPECIFIED)
00356 #endif
00357
00358 #ifdef DOXYGEN_SHOULD_SKIP_THIS
00359
00366     uint8_t emberAfRf4ceMsoIrRfDatabaseEntryGetMinimumNumberOfTransmissions
00367     (const EmberAfRf4ceMsoIrRfDatabaseRfDescriptor
00368         *rfDescriptor);
00367 #else
00368     #define emberAfRf4ceMsoIrRfDatabaseEntryGetMinimumNumberOfTransmissions(rfDescriptor) \
00369         READBITS((rfDescriptor)->rfConfig,
00369             \

```

```

00370     EMBER_AF_RF4CE_MSO_IR_RF_DATABASE_RF_CONFIG_MINIMUM_NUMBER_OF_TRANSMISSIONS_MASK)
00371 #endif
00372
00373 #ifdef DOXYGEN_SHOULD_SKIP_THIS
00374
00383     bool emberAfRf4ceMsoIrRfDatabaseEntryShouldTransmitUntilRelease
00384     (const EmberAfRf4ceMsoIrRfDatabaseRfDescriptor
00385      *rfDescriptor);
00384 #else
00385     #define
00386     emberAfRf4ceMsoIrRfDatabaseEntryShouldTransmitUntilRelease(rfDescriptor) \
00386     (READBITS((rfDescriptor)->rfConfig,
00387             \
00387     EMBER_AF_RF4CE_MSO_IR_RF_DATABASE_RF_CONFIG_KEEP_TRANSMITTING_UNTIL_KEY_RELEASE) \
00388     == \
00389     EMBER_AF_RF4CE_MSO_IR_RF_DATABASE_RF_CONFIG_KEEP_TRANSMITTING_UNTIL_KEY_RELEASE)
00389 #endif
00390
00391 #ifdef DOXYGEN_SHOULD_SKIP_THIS
00392
00399     bool emberAfRf4ceMsoIrRfDatabaseEntryHasIrDescriptor
00400     (const EmberAfRf4ceMsoIrRfDatabaseEntry *entry);
00400 #else
00401     #define emberAfRf4ceMsoIrRfDatabaseEntryHasIrDescriptor(entry) \
00402     (READBITS((entry)->flags,
00403             (EMBER_AF_RF4CE_MSO_IR_RF_DATABASE_FLAG_IR_SPECIFIED \
00404             | EMBER_AF_RF4CE_MSO_IR_RF_DATABASE_FLAG_USE_DEFAULT)) \
00405             == EMBER_AF_RF4CE_MSO_IR_RF_DATABASE_FLAG_IR_SPECIFIED)
00406 #endif
00407
00408 #endif // __RF4CE_MSO_H__
00409
00410 // END addtogroup

```

8.42 rf4ce-profile-internal.h File Reference

Data Structures

- struct [EmAfRf4cePowerSavingState](#)

Macros

- #define NWKC_MIN_CONTROLLER_PAIRING_TABLE_SIZE
- #define NWKC_MIN_TARGET_PAIRING_TABLE_SIZE
- #define emAfRf4ceStart
- #define emAfRf4ceSetPowerSavingParameters
- #define emAfRf4ceSetFrequencyAgilityParameters
- #define emAfRf4ceSetDiscoveryLqiThreshold
- #define emAfRf4ceGetBaseChannel
- #define emAfRf4ceDiscovery
- #define emAfRf4ceEnableAutoDiscoveryResponse
- #define emAfRf4cePair
- #define emAfRf4ceSetPairingTableEntry
- #define emAfRf4ceGetPairingTableEntry
- #define emAfRf4ceSetApplicationInfo
- #define emAfRf4ceGetApplicationInfo
- #define emAfRf4ceKeyUpdate
- #define emAfRf4ceSend
- #define emAfRf4ceUnpair
- #define emAfRf4ceStop
- #define emAfRf4ceGetMaxPayload

Functions

- bool `emAfRf4ceDiscoveryRequestHandler` (EmberEUI64 srcIeeeAddr, uint8_t nodeCapabilities, const EmberRf4ceVendorInfo *vendorInfo, const EmberRf4ceApplicationInfo *appInfo, uint8_t searchDevType, uint8_t rxLinkQuality)
- bool `emAfRf4ceDiscoveryResponseHandler` (bool atCapacity, uint8_t channel, EmberPanId panId, EmberEUI64 srcIeeeAddr, uint8_t nodeCapabilities, const EmberRf4ceVendorInfo *vendorInfo, const EmberRf4ceApplicationInfo *appInfo, uint8_t rxLinkQuality, uint8_t discRequestLqi)
- void `emAfRf4ceDiscoveryCompleteHandler` (EmberStatus status)
- void `emAfRf4ceAutoDiscoveryResponseCompleteHandler` (EmberStatus status, EmberEUI64 srcIeeeAddr, uint8_t nodeCapabilities, const EmberRf4ceVendorInfo *vendorInfo, const EmberRf4ceApplicationInfo *appInfo, uint8_t searchDevType)
- bool `emAfRf4cePairRequestHandler` (EmberStatus status, uint8_t pairingIndex, EmberEUI64 srcIeeeAddr, uint8_t nodeCapabilities, const EmberRf4ceVendorInfo *vendorInfo, const EmberRf4ceApplicationInfo *appInfo, uint8_t keyExchangeTransferCount)
- void `emAfRf4cePairCompleteHandler` (EmberStatus status, uint8_t pairingIndex, const EmberRf4ceVendorInfo *vendorInfo, const EmberRf4ceApplicationInfo *appInfo)
- void `emAfRf4ceMessageSentHandler` (EmberStatus status, uint8_t pairingIndex, uint8_t txOptions, uint8_t profileId, uint16_t vendorId, uint8_t messageTag, uint8_t messageLength, const uint8_t *message)
- void `emAfRf4ceIncomingMessageHandler` (uint8_t pairingIndex, uint8_t profileId, uint16_t vendorId, EmberRf4ceTxOption txOptions, uint8_t messageLength, const uint8_t *message)
- void `emAfRf4ceUnpairHandler` (uint8_t pairingIndex)
- void `emAfRf4ceUnpairCompleteHandler` (uint8_t pairingIndex)
- uint8_t `emAfRf4ceGetBaseChannel` (void)

Variables

- PGM `EmberAfRf4ceProfileId emAfRf4ceProfileIds []`
- bool `emAfRf4ceRxOnWhenIdleProfileStates []`
- `EmAfRf4cePowerSavingState emAfRf4cePowerSavingState`

8.42.1 Macro Definition Documentation

8.42.1.1 #define NWKC_MIN_CONTROLLER_PAIRING_TABLE_SIZE

Definition at line 3 of file `rf4ce-profile-internal.h`.

8.42.1.2 #define NWKC_MIN_TARGET_PAIRING_TABLE_SIZE

Definition at line 4 of file `rf4ce-profile-internal.h`.

8.42.1.3 #define emAfRf4ceStart

Definition at line 89 of file `rf4ce-profile-internal.h`.

8.42.1.4 #define emAfRf4ceSetPowerSavingParameters

Definition at line 90 of file `rf4ce-profile-internal.h`.

8.42.1.5 #define emAfRf4ceSetFrequencyAgilityParameters

Definition at line 91 of file [rf4ce-profile-internal.h](#).

8.42.1.6 #define emAfRf4ceSetDiscoveryLqiThreshold

Definition at line 92 of file [rf4ce-profile-internal.h](#).

8.42.1.7 #define emAfRf4ceGetBaseChannel

Definition at line 93 of file [rf4ce-profile-internal.h](#).

8.42.1.8 #define emAfRf4ceDiscovery

Definition at line 94 of file [rf4ce-profile-internal.h](#).

8.42.1.9 #define emAfRf4ceEnableAutoDiscoveryResponse

Definition at line 95 of file [rf4ce-profile-internal.h](#).

8.42.1.10 #define emAfRf4cePair

Definition at line 96 of file [rf4ce-profile-internal.h](#).

8.42.1.11 #define emAfRf4ceSetPairingTableEntry

Definition at line 97 of file [rf4ce-profile-internal.h](#).

8.42.1.12 #define emAfRf4ceGetPairingTableEntry

Definition at line 98 of file [rf4ce-profile-internal.h](#).

8.42.1.13 #define emAfRf4ceSetApplicationInfo

Definition at line 99 of file [rf4ce-profile-internal.h](#).

8.42.1.14 #define emAfRf4ceGetApplicationInfo

Definition at line 100 of file [rf4ce-profile-internal.h](#).

8.42.1.15 #define emAfRf4ceKeyUpdate

Definition at line 101 of file [rf4ce-profile-internal.h](#).

8.42.1.16 #define emAfRf4ceSend

Definition at line 102 of file [rf4ce-profile-internal.h](#).

8.42.1.17 #define emAfRf4ceUnpair

Definition at line 103 of file [rf4ce-profile-internal.h](#).

8.42.1.18 #define emAfRf4ceStop

Definition at line 104 of file [rf4ce-profile-internal.h](#).

8.42.1.19 #define emAfRf4ceGetMaxPayload

Definition at line 105 of file [rf4ce-profile-internal.h](#).

8.42.2 Function Documentation

- 8.42.2.1 `bool emAfRf4ceDiscoveryRequestHandler (EmberEUI64 srcIEEEAddr, uint8_t nodeCapabilities, const EmberRf4ceVendorInfo * vendorInfo, const EmberRf4ceApplicationInfo * appInfo, uint8_t searchDevType, uint8_t rxLinkQuality)`
- 8.42.2.2 `bool emAfRf4ceDiscoveryResponseHandler (bool atCapacity, uint8_t channel, EmberPanId panId, EmberEUI64 srcIEEEAddr, uint8_t nodeCapabilities, const EmberRf4ceVendorInfo * vendorInfo, const EmberRf4ceApplicationInfo * appInfo, uint8_t rxLinkQuality, uint8_t discRequestLqi)`
- 8.42.2.3 `void emAfRf4ceDiscoveryCompleteHandler (EmberStatus status)`
- 8.42.2.4 `void emAfRf4ceAutoDiscoveryResponseCompleteHandler (EmberStatus status, EmberEUI64 srcIEEEAddr, uint8_t nodeCapabilities, const EmberRf4ceVendorInfo * vendorInfo, const EmberRf4ceApplicationInfo * appInfo, uint8_t searchDevType)`
- 8.42.2.5 `bool emAfRf4cePairRequestHandler (EmberStatus status, uint8_t pairingIndex, EmberEUI64 srcIEEEAddr, uint8_t nodeCapabilities, const EmberRf4ceVendorInfo * vendorInfo, const EmberRf4ceApplicationInfo * appInfo, uint8_t keyExchangeTransferCount)`
- 8.42.2.6 `void emAfRf4cePairCompleteHandler (EmberStatus status, uint8_t pairingIndex, const EmberRf4ceVendorInfo * vendorInfo, const EmberRf4ceApplicationInfo * appInfo)`
- 8.42.2.7 `void emAfRf4ceMessageSentHandler (EmberStatus status, uint8_t pairingIndex, uint8_t txOptions, uint8_t profileId, uint16_t vendorId, uint8_t messageTag, uint8_t messageLength, const uint8_t * message)`
- 8.42.2.8 `void emAfRf4ceIncomingMessageHandler (uint8_t pairingIndex, uint8_t profileId, uint16_t vendorId, EmberRf4ceTxOption txOptions, uint8_t messageLength, const uint8_t * message)`
- 8.42.2.9 `void emAfRf4ceUnpairHandler (uint8_t pairingIndex)`
- 8.42.2.10 `void emAfRf4ceUnpairCompleteHandler (uint8_t pairingIndex)`

8.42.2.11 uint8_t emAfRf4ceGetBaseChannel (void)

8.42.3 Variable Documentation

8.42.3.1 PGM EmberAfRf4ceProfileId emAfRf4ceProfileIds[]

8.42.3.2 bool emAfRf4ceRxOnWhenIdleProfileStates[]

8.42.3.3 EmAfRf4cePowerSavingState emAfRf4cePowerSavingState

8.43 rf4ce-profile-internal.h

```

00001 // Copyright 2013 Silicon Laboratories, Inc.
00002
00003 #define NWKC_MIN_CONTROLLER_PAIRING_TABLE_SIZE 1
00004 #define NWKC_MIN_TARGET_PAIRING_TABLE_SIZE      5
00005
00006 bool emAfRf4ceDiscoveryRequestHandler(
00007     EmberEUI64 srcIeeeAddr,
00008                                         uint8_t nodeCapabilities,
00009                                         const EmberRf4ceVendorInfo *vendorInfo
00010     ,
00011     const EmberRf4ceApplicationInfo *
00012     appInfo,
00013     uint8_t searchDevType,
00014     uint8_t rxLinkQuality);
00015
00016 bool emAfRf4ceDiscoveryResponseHandler(bool
00017     atCapacity,
00018     uint8_t channel,
00019     EmberPanId panId,
00020     EmberEUI64 srcIeeeAddr,
00021     uint8_t nodeCapabilities,
00022     const EmberRf4ceVendorInfo *
00023     vendorInfo,
00024     const EmberRf4ceApplicationInfo *
00025     appInfo,
00026     uint8_t rxLinkQuality,
00027     uint8_t discRequestLqi);
00028
00029 void emAfRf4ceDiscoveryCompleteHandler(
00030     EmberStatus status);
00031
00032 void emAfRf4ceAutoDiscoveryResponseCompleteHandler
00033     (EmberStatus status,
00034     EmberEUI64 srcIeeeAddr,
00035     uint8_t nodeCapabilities,
00036     const EmberRf4ceVendorInfo *vendorInfo,
00037     const EmberRf4ceApplicationInfo *appInfo,
00038     uint8_t keyExchangeTransferCount);
00039
00040 void emAfRf4cePairCompleteHandler(EmberStatus
00041     status,
00042     uint8_t pairingIndex,
00043     const EmberRf4ceVendorInfo *vendorInfo,
00044     const EmberRf4ceApplicationInfo *appInfo);
00045
00046 void emAfRf4ceMessageSentHandler(EmberStatus status,
00047     uint8_t pairingIndex,
00048     uint8_t txOptions,
00049     uint8_t profileId,
00050     uint16_t vendorId,
00051     uint8_t messageTag,
```

```

00051         uint8_t messageLength,
00052         const uint8_t *message);
00053
00054 void emAfRf4ceIncomingMessageHandler(uint8_t
00055     pairingIndex,
00056             uint8_t profileId,
00057             uint16_t vendorId,
00058             EmberRf4ceTxOption txOptions,
00059             uint8_t messageLength,
00060             const uint8_t *message);
00061
00061 void emAfRf4ceUnpairHandler(uint8_t pairingIndex);
00062
00063 void emAfRf4ceUnpairCompleteHandler(uint8_t
00064     pairingIndex);
00065 uint8_t emAfRf4ceGetBaseChannel(void);
00066
00067 #ifdef EZSP_HOST
00068     #define emAfRf4ceStart                      ezspRf4ceStart
00069     #define emAfRf4ceSetPowerSavingParameters      ezspRf4ceSetPowerSavingParameters
00070     #define emAfRf4ceSetFrequencyAgilityParameters ezspRf4ceSetFrequencyAgilityParameters
00071     #define emAfRf4ceSetDiscoveryLqiThreshold(threshold)
00072         ezspSetValue(EZSP_VALUE_RF4CE_DISCOVERY_LQI_THRESHOLD, 1, &(threshold)) \
00073 // emAfRf4ceGetBaseChannel is a function defined in rf4ce-profile-host.c
00074     #define emAfRf4ceDiscovery                  ezspRf4ceDiscovery
00075     #define emAfRf4ceEnableAutoDiscoveryResponse   ezspRf4ceEnableAutoDiscoveryResponse
00076     #define emAfRf4cePair                        ezspRf4cePair
00077     #define emAfRf4ceSetPairingTableEntry(pairingIndex, entry)
00078     \
00079     (entry == NULL) ? ezspRf4ceDeletePairingTableEntry(pairingIndex)
00080     \
00081     : ezspRf4ceSetPairingTableEntry(pairingIndex, entry)
00082     #define emAfRf4ceGetPairingTableEntry        ezspRf4ceGetPairingTableEntry
00083     #define emAfRf4ceSetApplicationInfo          ezspRf4ceSetApplicationInfo
00084     #define emAfRf4ceGetApplicationInfo          ezspRf4ceGetApplicationInfo
00085     #define emAfRf4ceKeyUpdate                  ezspRf4ceKeyUpdate
00086     #define emAfRf4ceSend                      ezspRf4ceSend
00087     #define emAfRf4ceUnpair                    ezspRf4ceUnpair
00088     #define emAfRf4ceStop                     ezspRf4ceStop
00089     #define emAfRf4ceGetMaxPayload            ezspRf4ceGetMaxPayload
00090     #else
00091     #define emAfRf4ceStart                  emberRf4ceStart
00092     #define emAfRf4ceSetPowerSavingParameters emberRf4ceSetPowerSavingParameters
00093     #define emAfRf4ceSetFrequencyAgilityParameters emberRf4ceSetFrequencyAgilityParameters
00094     #define emAfRf4ceSetDiscoveryLqiThreshold emberRf4ceSetDiscoveryLqiThreshold
00095     #define emAfRf4ceGetBaseChannel          emberRf4ceGetBaseChannel
00096     #define emAfRf4ceDiscovery            emberRf4ceDiscovery
00097     #define emAfRf4ceEnableAutoDiscoveryResponse emberRf4ceEnableAutoDiscoveryResponse
00098     #define emAfRf4cePair                emberRf4cePair
00099     #define emAfRf4ceSetPairingTableEntry  emberRf4ceSetPairingTableEntry
00100    #define emAfRf4ceGetPairingTableEntry  emberRf4ceGetPairingTableEntry
00101    #define emAfRf4ceSetApplicationInfo  emberRf4ceSetApplicationInfo
00102    #define emAfRf4ceGetApplicationInfo  emberRf4ceGetApplicationInfo
00103    #define emAfRf4ceKeyUpdate          emberRf4ceKeyUpdate
00104    #define emAfRf4ceSend              emberRf4ceSend
00105    #define emAfRf4ceUnpair            emberRf4ceUnpair
00106    #define emAfRf4ceStop              emberRf4ceStop
00107    #define emAfRf4ceGetMaxPayload    emberRf4ceGetMaxPayload
00108    #endif
00109
00110
00111 typedef struct {
00112     uint32_t dutyCycleMs;
00113     uint32_t activePeriodMs;
00114 } EmAfRf4cePowerSavingState;
00115
00116 extern PGM EmberAfRf4ceProfileId emAfRf4ceProfileIds
00117 [];
00118 extern bool emAfRf4ceRxOnWhenIdleProfileStates
00119 [];
00120 extern EmAfRf4cePowerSavingState
00121 emAfRf4cePowerSavingState;

```

8.44 rf4ce-profile-types.h File Reference

TypeDefs

- `typedef void(* EmberAfRf4cePairCompleteCallback)(uint8_t, EmberStatus, const EmberRf4ceVendorInfo *, const EmberRf4ceApplicationInfo *)`

Enumerations

- `enum EmberAfRf4ceStatus {
 EMBER_AF_RF4CE_STATUS_SUCCESS, EMBER_AF_RF4CE_STATUS_NO_ORG_CAPACITY,
 EMBER_AF_RF4CE_STATUS_NO_REC_CAPACITY, EMBER_AF_RF4CE_STATUS_NO_PAIRING,
 EMBER_AF_RF4CE_STATUS_NO_RESPONSE, EMBER_AF_RF4CE_STATUS_NOT_PERMITTED,
 EMBER_AF_RF4CE_STATUS_DUPLICATE_PAIRING, EMBER_AF_RF4CE_STATUS_FRAME_COUNTER_EXPIRED,
 EMBER_AF_RF4CE_STATUS_DISCOVERY_ERROR, EMBER_AF_RF4CE_STATUS_DISCOVERY_TIMEOUT,
 EMBER_AF_RF4CE_STATUS_SECURITY_FAILURE,
 EMBER_AF_RF4CE_STATUS_INVALID_PARAMETER, EMBER_AF_RF4CE_STATUS_UNSUPPORTED_ATTRIBUTE,
 EMBER_AF_RF4CE_STATUS_INVALID_INDEX }`
- `enum EmberAfRf4ceProfileId {
 EMBER_AF_RF4CE_PROFILE_GENERIC_DEVICE, EMBER_AF_RF4CE_PROFILE_CONSUMER_ELECTRONICS_REMOTE_CONTROL, EMBER_AF_RF4CE_PROFILE_REMOTE_CONTROL_1_0,
 EMBER_AF_RF4CE_PROFILE_REMOTE_CONTROL_1_1, EMBER_AF_RF4CE_PROFILE_INPUT_DEVICE_1_0, EMBER_AF_RF4CE_PROFILE_REMOTE_CONTROL_2_0,
 EMBER_AF_RF4CE_PROFILE_MSO, EMBER_AF_RF4CE_PROFILE_WILDCARD }`
- `enum EmberAfRf4ceVendor {
 EMBER_AF_RF4CE_VENDOR_PANASONIC, EMBER_AF_RF4CE_VENDOR_SONY, EMBER_AF_RF4CE_VENDOR_SAMSUNG,
 EMBER_AF_RF4CE_VENDOR_PHILIPS, EMBER_AF_RF4CE_VENDOR_FREESCALE, EMBER_AF_RF4CE_VENDOR_OKI_SEMICONDUCTOR,
 EMBER_AF_RF4CE_VENDOR_TEST_VENDOR_1, EMBER_AF_RF4CE_VENDOR_TEST_VENDOR_2, EMBER_AF_RF4CE_VENDOR_TEST_VENDOR_3 }`
- `enum EmberAfRf4ceDeviceType {
 EMBER_AF_RF4CE_DEVICE_TYPE_RESERVED, EMBER_AF_RF4CE_DEVICE_TYPE_REMOTE_CONTROL,
 EMBER_AF_RF4CE_DEVICE_TYPE_TELEVISION, EMBER_AF_RF4CE_DEVICE_TYPE_PROJECTOR,
 EMBER_AF_RF4CE_DEVICE_TYPE_PLAYER, EMBER_AF_RF4CE_DEVICE_TYPE_RECORDER,
 EMBER_AF_RF4CE_DEVICE_TYPE_VIDEO_PLAYER_RECORDER, EMBER_AF_RF4CE_DEVICE_TYPE_AUDIO_PLAYER_RECORDER,
 EMBER_AF_RF4CE_DEVICE_TYPE_AUDIO_VIDEO_RECORDER, EMBER_AF_RF4CE_DEVICE_TYPE_SET_TOP_BOX,
 EMBER_AF_RF4CE_DEVICE_TYPE_HOME_THEATER_SYSTEM, EMBER_AF_RF4CE_DEVICE_TYPE_MEDIA_CENTER_PC,
 EMBER_AF_RF4CE_DEVICE_TYPE_GAME_CONSOLE, EMBER_AF_RF4CE_DEVICE_TYPE_SATELLITE_RADIO_RECEIVER,
 EMBER_AF_RF4CE_DEVICE_TYPE_IR_EXTENDER, EMBER_AF_RF4CE_DEVICE_TYPE_MONITOR,
 EMBER_AF_RF4CE_DEVICE_TYPE_GENERIC, EMBER_AF_RF4CE_DEVICE_TYPE_WILDCARD }`

8.44.1 Typedef Documentation

8.44.1.1 `typedef void(* EmberAfRf4cePairCompleteCallback)(uint8_t, EmberStatus, const EmberRf4ceVendorInfo *, const EmberRf4ceApplicationInfo *)`

Function type definition of the pair complete callback.

Definition at line 123 of file [rf4ce-profile-types.h](#).

8.44.2 Enumeration Type Documentation

8.44.2.1 enum EmberAfRf4ceStatus

ZigBee RF4CE status.

Enumerator:

```
EMBER_AF_RF4CE_STATUS_SUCCESS
EMBER_AF_RF4CE_STATUS_NO_ORG_CAPACITY
EMBER_AF_RF4CE_STATUS_NO_REC_CAPACITY
EMBER_AF_RF4CE_STATUS_NO_PAIRING
EMBER_AF_RF4CE_STATUS_NO_RESPONSE
EMBER_AF_RF4CE_STATUS_NOT_PERMITTED
EMBER_AF_RF4CE_STATUS_DUPLICATE_PAIRING
EMBER_AF_RF4CE_STATUS_FRAME_COUNTER_EXPIRED
EMBER_AF_RF4CE_STATUS_DISCOVERY_ERROR
EMBER_AF_RF4CE_STATUS_DISCOVERY_TIMEOUT
EMBER_AF_RF4CE_STATUS_SECURITY_TIMEOUT
EMBER_AF_RF4CE_STATUS_SECURITY_FAILURE
EMBER_AF_RF4CE_STATUS_INVALID_PARAMETER
EMBER_AF_RF4CE_STATUS_UNSUPPORTED_ATTRIBUTE
EMBER_AF_RF4CE_STATUS_INVALID_INDEX
```

Definition at line 10 of file [rf4ce-profile-types.h](#).

8.44.2.2 enum EmberAfRf4ceProfileId

ZigBee RF4CE profile identifier.

Enumerator:

EMBER_AF_RF4CE_PROFILE_GENERIC_DEVICE Generic Device Profile (GDP) versions 1.-0 and 2.0.

EMBER_AF_RF4CE_PROFILE_CONSUMER_ELECTRONICS_REMOTE_CONTROL The Consumer Electronics Remote Control (CERC) profile was renamed to the ZigBee Remote Control (ZRC) profile when the specification went from version 1.0 in document 09-4946-00 to version 1.1 in document 09-4946-01. 1.1 is backwards compatible with 1.0. For convenience, the profile can be referred to as CERC, ZRC 1.0, or ZRC 1.1.

EMBER_AF_RF4CE_PROFILE_REMOTE_CONTROL_1_0 A convenience alias for [EMBER_AF_RF4CE_PROFILE_CONSUMER_ELECTRONICS_REMOTE_CONTROL](#).

EMBER_AF_RF4CE_PROFILE_REMOTE_CONTROL_1_1 ZigBee Remote Control (ZRC) profile version 1.1.

EMBER_AF_RF4CE_PROFILE_INPUT_DEVICE_1_0 ZigBee Input Device (ZID) profile version 1.0.

EMBER_AF_RF4CE_PROFILE_REMOTE_CONTROL_2_0 ZigBee Remote Control (ZRC) profile version 2.0.

EMBER_AF_RF4CE_PROFILE_MSO Multiple System Operators (MSO) profile.

EMBER_AF_RF4CE_PROFILE_WILDCARD Wildcard profile.

Definition at line [37](#) of file [rf4ce-profile-types.h](#).

8.44.2.3 enum EmberAfRf4ceVendor

ZigBee RF4CE vendor identifier.

Enumerator:

```
EMBER_AF_RF4CE_VENDOR_PANASONIC
EMBER_AF_RF4CE_VENDOR_SONY
EMBER_AF_RF4CE_VENDOR_SAMSUNG
EMBER_AF_RF4CE_VENDOR_PHILIPS
EMBER_AF_RF4CE_VENDOR_FREESCALE
EMBER_AF_RF4CE_VENDOR_OKI_SEMICONDUCTOR
EMBER_AF_RF4CE_VENDOR_TEXAS_INSTRUMENTS
EMBER_AF_RF4CE_VENDOR_TEST_VENDOR_1
EMBER_AF_RF4CE_VENDOR_TEST_VENDOR_2
EMBER_AF_RF4CE_VENDOR_TEST_VENDOR_3
```

Definition at line [72](#) of file [rf4ce-profile-types.h](#).

8.44.2.4 enum EmberAfRf4ceDeviceType

ZigBee RF4CE device type.

Enumerator:

```
EMBER_AF_RF4CE_DEVICE_TYPE_RESERVED
EMBER_AF_RF4CE_DEVICE_TYPE_REMOTE_CONTROL
EMBER_AF_RF4CE_DEVICE_TYPE_TELEVISION
EMBER_AF_RF4CE_DEVICE_TYPE_PROJECTOR
EMBER_AF_RF4CE_DEVICE_TYPE_PLAYER
EMBER_AF_RF4CE_DEVICE_TYPE_RECORDER
EMBER_AF_RF4CE_DEVICE_TYPE_VIDEO_PLAYER_RECORDER
EMBER_AF_RF4CE_DEVICE_TYPE_AUDIO_PLAYER_RECORDER
```

EMBER_AF_RF4CE_DEVICE_TYPE_AUDIO_VIDEO_RECORDER
EMBER_AF_RF4CE_DEVICE_TYPE_SET_TOP_BOX
EMBER_AF_RF4CE_DEVICE_TYPE_HOME_THEATER_SYSTEM
EMBER_AF_RF4CE_DEVICE_TYPE_MEDIA_CENTER_PC
EMBER_AF_RF4CE_DEVICE_TYPE_GAME_CONSOLE
EMBER_AF_RF4CE_DEVICE_TYPE_SATELLITE_RADIO_RECEIVER
EMBER_AF_RF4CE_DEVICE_TYPE_IR_EXTENDER
EMBER_AF_RF4CE_DEVICE_TYPE_MONITOR
EMBER_AF_RF4CE_DEVICE_TYPE_GENERIC
EMBER_AF_RF4CE_DEVICE_TYPE_WILDCARD

Definition at line 94 of file [rf4ce-profile-types.h](#).

8.45 rf4ce-profile-types.h

```
00001 // Copyright 2014 Silicon Laboratories, Inc.
00002
00003 #ifndef __RF4CE_PROFILE_TYPES_H__
00004 #define __RF4CE_PROFILE_TYPES_H__
00005
00009 #ifdef DOXYGEN_SHOULD_SKIP_THIS
00010 enum EmberAfRf4ceStatus
00011 {
00012     #else
00012     typedef uint8_t EmberAfRf4ceStatus;
00013     enum
00014     {
00015         #endif
00016         EMBER_AF_RF4CE_STATUS_SUCCESS = 0x00,
00017         EMBER_AF_RF4CE_STATUS_NO_ORG_CAPACITY = 0xB0,
00018         EMBER_AF_RF4CE_STATUS_NO_REC_CAPACITY = 0xB1,
00019         EMBER_AF_RF4CE_STATUS_NO_PAIRING = 0xB2,
00020         EMBER_AF_RF4CE_STATUS_NO_RESPONSE = 0xB3,
00021         EMBER_AF_RF4CE_STATUS_NOT_PERMITTED = 0xB4,
00022         EMBER_AF_RF4CE_STATUS_DUPLICATE_PAIRING = 0xB5,
00023         EMBER_AF_RF4CE_STATUS_FRAME_COUNTER_EXPIRED = 0xB6,
00024         EMBER_AF_RF4CE_STATUS_DISCOVERY_ERROR = 0xB7,
00025         EMBER_AF_RF4CE_STATUS_DISCOVERY_TIMEOUT = 0xB8,
00026         EMBER_AF_RF4CE_STATUS_SECURITY_TIMEOUT = 0xB9,
00027         EMBER_AF_RF4CE_STATUS_SECURITY_FAILURE = 0xBA,
00028         EMBER_AF_RF4CE_STATUS_INVALID_PARAMETER = 0xE8,
00029         EMBER_AF_RF4CE_STATUS_UNSUPPORTED_ATTRIBUTE = 0xF4,
00030         EMBER_AF_RF4CE_STATUS_INVALID_INDEX = 0xF9,
00031     };
00032
00036 #ifdef DOXYGEN_SHOULD_SKIP_THIS
00037 enum EmberAfRf4ceProfileId
00038 {
00039     #else
00039     typedef uint8_t EmberAfRf4ceProfileId;
00040     enum
00041     {
00042 }
```

```

00044 EMBER_AF_RF4CE_PROFILE_GENERIC_DEVICE
00045     = 0x00,
00051 EMBER_AF_RF4CE_PROFILE_CONSUMER_ELECTRONICS_REMOTE_CONTROL
00052     = 0x01,
00055 EMBER_AF_RF4CE_PROFILE_REMOTE_CONTROL_1_0
00056     = 0x01,
00057 EMBER_AF_RF4CE_PROFILE_REMOTE_CONTROL_1_1
00058     = 0x01,
00059 EMBER_AF_RF4CE_PROFILE_INPUT_DEVICE_1_0
00060     = 0x02,
00061 EMBER_AF_RF4CE_PROFILE_REMOTE_CONTROL_2_0
00062     = 0x03,
00063 EMBER_AF_RF4CE_PROFILE_MSO
00064     = 0xC0,
00065 EMBER_AF_RF4CE_PROFILE_WILDCARD
00066     = 0xFF,
00067 };
00068 #ifndef DOXYGEN_SHOULD_SKIP_THIS
00069 enum EmberAfRf4ceVendor
00070 {
00071 #else
00072 typedef uint16_t EmberAfRf4ceVendor;
00073 enum
00074 #endif
00075 {
00076     EMBER_AF_RF4CE_VENDOR_PANASONIC      =
00077         0x0001,
00078     EMBER_AF_RF4CE_VENDOR_SONY           =
00079         0x0002,
00080     EMBER_AF_RF4CE_VENDOR_SAMSUNG        =
00081         0x0003
00082     ,
00083     EMBER_AF_RF4CE_VENDOR_PHILIPS        =
00084         0x0004
00085     ,
00086     EMBER_AF_RF4CE_VENDOR_FREESCALE       =
00087         0x0005,
00088     EMBER_AF_RF4CE_VENDOR_OKI_SEMICONDUCTOR
00089         = 0x0006,
00090     EMBER_AF_RF4CE_VENDOR_TEXAS_INSTRUMENTS
00091         = 0x0007,
00092     EMBER_AF_RF4CE_VENDOR_TEST_VENDOR_1   =
00093         0xFFFF1,
00094     EMBER_AF_RF4CE_VENDOR_TEST_VENDOR_2   =
00095         0xFFFF2,
00096     EMBER_AF_RF4CE_VENDOR_TEST_VENDOR_3   =
00097         0xFFFF3,
00098 };
00099 #endif DOXYGEN_SHOULD_SKIP_THIS
00100 enum EmberAfRf4ceDeviceType
00101 {
00102 #else
00103 typedef uint8_t EmberAfRf4ceDeviceType;
00104 enum
00105 #endif
00106 {
00107     EMBER_AF_RF4CE_DEVICE_TYPE_RESERVED
00108         = 0x00,
00109     EMBER_AF_RF4CE_DEVICE_TYPE_REMOTE_CONTROL
00110         = 0x01,
00111     EMBER_AF_RF4CE_DEVICE_TYPE_TELEVISION
00112         = 0x02,
00113     EMBER_AF_RF4CE_DEVICE_TYPE_PROJECTOR
00114         = 0x03,
00115     EMBER_AF_RF4CE_DEVICE_TYPE_PLAYER
00116         = 0x04,
00117     EMBER_AF_RF4CE_DEVICE_TYPE_RECORDER
00118         = 0x05,
00119     EMBER_AF_RF4CE_DEVICE_TYPE_VIDEO_PLAYER_RECORDER
00120         = 0x06,
00121     EMBER_AF_RF4CE_DEVICE_TYPE_AUDIO_PLAYER_RECORDER
00122         = 0x07,
00123     EMBER_AF_RF4CE_DEVICE_TYPE_AUDIO_VIDEO_RECORDER
00124         = 0x08,
00125     EMBER_AF_RF4CE_DEVICE_TYPE_SET_TOP_BOX
00126         = 0x09,
00127     EMBER_AF_RF4CE_DEVICE_TYPE_HOME_THEATER_SYSTEM
00128         = 0x0A,
00129     EMBER_AF_RF4CE_DEVICE_TYPE_MEDIA_CENTER_PC
00130         = 0x0B,
00131     EMBER_AF_RF4CE_DEVICE_TYPE_GAME_CONSOLE
00132         = 0x0C,
00133     EMBER_AF_RF4CE_DEVICE_TYPE_SATELLITE_RADIO_RECEIVER

```

```

00114     = 0x0D,
00114     EMBER_AF_RF4CE_DEVICE_TYPE_IR_EXTENDER
00115         = 0x0E,
00115     EMBER_AF_RF4CE_DEVICE_TYPE_MONITOR
00116         = 0x0F,
00116     EMBER_AF_RF4CE_DEVICE_TYPE_GENERIC
00117         = 0xFE,
00117     EMBER_AF_RF4CE_DEVICE_TYPE_WILDCARD
00118         = 0xFF,
00118 };
00119
00123 typedef void(*EmberAfRf4cePairCompleteCallback)
00123     (uint8_t,
00124         EmberStatus,
00125         const EmberRf4ceVendorInfo *,
00126         const EmberRf4ceApplicationInfo
00126     *);
00127
00128 #endif // __RF4CE_PROFILE_TYPES_H__

```

8.46 rf4ce-profile.h File Reference

```
#include "rf4ce-profile-types.h"
```

Macros

- #define EMBER_AF_RF4CE_MAXIMUM_RF4CE_PAYLOAD_LENGTH
- #define EMBER_AF_RF4CE_MESSAGE_TAG_MASK

Functions

- bool emberAfRf4ceIsCurrentNetwork (void)
- EmberStatus emberAfRf4cePushNetworkIndex (void)
- EmberStatus emberAfRf4ceStart (void)
- EmberStatus emberAfRf4ceSetPowerSavingParameters (uint32_t dutyCycleMs, uint32_t activePeriodMs)
- EmberStatus emberAfRf4ceRxEnable (EmberAfRf4ceProfileId profileId, bool enable)
- EmberStatus emberAfRf4ceSetFrequencyAgilityParameters (uint8_t rssiWindowSize, uint8_t channelChangeReads, int8_t rssithreshold, uint16_t readIntervalS, uint8_t readDuration)
- EmberStatus emberAfRf4ceSetDiscoveryLqiThreshold (uint8_t threshold)
- uint8_t emberAfRf4ceGetBaseChannel (void)
- EmberStatus emberAfRf4ceDiscovery (EmberPanId panId, EmberNodeId nodeId, uint8_t searchDevType, uint16_t discDurationMs, uint8_t maxDiscRepetitions, uint8_t discProfileIdListLength, uint8_t *discProfileIdList)
- EmberStatus emberAfRf4ceEnableAutoDiscoveryResponse (uint16_t durationMs, uint8_t discProfileIdListLength, uint8_t *discProfileIdList)
- EmberStatus emberAfRf4cePair (uint8_t channel, EmberPanId panId, EmberEUI64 ieeeAddr, uint8_t keyExchangeTransferCount, EmberAfRf4cePairCompleteCallback pairCompleteCallback)
- uint8_t emberAfRf4ceGetPairingIndex (void)
- EmberStatus emberAfRf4ceSetPairingTableEntry (uint8_t pairingIndex, EmberRf4cePairingTableEntry *entry)
- EmberStatus emberAfRf4ceGetPairingTableEntry (uint8_t pairingIndex, EmberRf4cePairingTableEntry *entry)
- EmberStatus emberAfRf4ceSetApplicationInfo (EmberRf4ceApplicationInfo *appInfo)

- EmberStatus `emberAfRf4ceGetApplicationInfo` (EmberRf4ceApplicationInfo *appInfo)
- EmberStatus `emberAfRf4ceKeyUpdate` (uint8_t pairingIndex, EmberKeyData *key)
- EmberStatus `emberAfRf4ceSend` (uint8_t pairingIndex, uint8_t profileId, uint8_t *message, uint8_t messageLength, uint8_t *messageTag)
- EmberStatus `emberAfRf4ceSendVendorSpecific` (uint8_t pairingIndex, uint8_t profileId, uint16_t vendorId, uint8_t *message, uint8_t messageLength, uint8_t *messageTag)
- EmberStatus `emberAfRf4ceSendExtended` (uint8_t pairingIndex, uint8_t profileId, uint16_t vendorId, EmberRf4ceTxOption txOptions, uint8_t *message, uint8_t messageLength, uint8_t *messageTag)
- EmberStatus `emberAfRf4ceGetDefaultTxOptions` (uint8_t pairingIndex, EmberRf4ceTxOption *txOptions)
- EmberStatus `emberAfRf4ceUnpair` (uint8_t pairingIndex)
- EmberStatus `emberAfRf4ceStop` (void)
- uint8_t `emberAfRf4ceGetMaxPayload` (uint8_t pairingIndex, EmberRf4ceTxOption txOptions)
- uint8_t `emberAfRf4ceDeviceTypeListLength` (EmberRf4ceApplicationCapabilities capabilities)
- uint8_t `emberAfRf4ceProfileIdListLength` (EmberRf4ceApplicationCapabilities capabilities)
- uint16_t `emberAfRf4ceVendorId` (void)
- bool `emberAfRf4ceIsDeviceTypeSupported` (const EmberRf4ceApplicationInfo *appInfo, **EmberAfRf4ceDeviceType** deviceType)
- bool `emberAfRf4ceIsDeviceTypeSupportedLocally` (**EmberAfRf4ceDeviceType** deviceType)
- bool `emberAfRf4ceIsProfileSupported` (const EmberRf4ceApplicationInfo *appInfo, **EmberAfRf4ceProfileId** profileId)
- bool `emberAfRf4ceIsProfileSupportedLocally` (**EmberAfRf4ceProfileId** profileId)
- bool `emberAfRf4cePairingTableEntryIsUnused` (const EmberRf4cePairingTableEntry *pairingTableEntry)
- bool `emberAfRf4cePairingTableEntryIsProvisional` (const EmberRf4cePairingTableEntry *pairingTableEntry)
- bool `emberAfRf4cePairingTableEntryIsActive` (const EmberRf4cePairingTableEntry *pairingTableEntry)
- bool `emberAfRf4cePairingTableEntryHasLinkKey` (const EmberRf4cePairingTableEntry *pairingTableEntry)
- bool `emberAfRf4cePairingTableEntryIsPairingInitiator` (const EmberRf4cePairingTableEntry *pairingTableEntry)
- bool `emberAfRf4cePairingTableEntryHasSecurity` (const EmberRf4cePairingTableEntry *pairingTableEntry)
- bool `emberAfRf4cePairingTableEntryHasChannelNormalization` (const EmberRf4cePairingTableEntry *pairingTableEntry)

8.47 rf4ce-profile.h

```

00001 // Copyright 2014 Silicon Laboratories, Inc.
00002
00003 #ifndef __RF4CE_PROFILE_H__
00004 #define __RF4CE_PROFILE_H__
00005
00006 #include "rf4ce-profile-types.h"
00007
00048 bool emberAfRf4ceIsCurrentNetwork(void);
00049
00063 EmberStatus emberAfRf4cePushNetworkIndex(void);
00064
00079 EmberStatus emberAfRf4ceStart(void);
00080
00100 EmberStatus emberAfRf4ceSetPowerSavingParameters
    (uint32_t dutyCycleMs,
     uint32_t activePeriodMs);
00101

```

```

00102
00125 EmberStatus emberAfRf4ceRxEnable(EmberAfRf4ceProfileId
00126     profileId,
00127             bool enable);
00127
00142 EmberStatus emberAfRf4ceSetFrequencyAgilityParameters
00143     (uint8_t rssiWindowSize,
00143         uint8_t
00144     channelChangeReads,
00144             int8_t rssiThreshold,
00145             uint16_t readIntervals,
00146             uint8_t readDuration);
00147
00162 EmberStatus emberAfRf4ceSetDiscoveryLqiThreshold
00163     (uint8_t threshold);
00164
00178 uint8_t emberAfRf4ceGetBaseChannel(void);
00179
00199 EmberStatus emberAfRf4ceDiscovery(EmberPanId panId,
00200     EmberNodeId nodeId,
00201             uint8_t searchDevType,
00202             uint16_t discDurationMs,
00203             uint8_t maxDiscRepetitions,
00204             uint8_t discProfileIdListLength,
00205             uint8_t *discProfileIdList);
00206
00223 EmberStatus emberAfRf4ceEnableAutoDiscoveryResponse
00224     (uint16_t durationMs,
00224         uint8_t
00225     discProfileIdListLength,
00225         uint8_t *discProfileIdList)
00226 ;
00226
00243 EmberStatus emberAfRf4cePair(uint8_t channel,
00244     EmberPanId panId,
00245     EmberEUI64 ieeeAddr,
00246     uint8_t keyExchangeTransferCount,
00247     EmberAfRf4cePairCompleteCallback
00248     pairCompleteCallback);
00248
00267 uint8_t emberAfRf4ceGetPairingIndex(void);
00268
00282 EmberStatus emberAfRf4ceSetPairingTableEntry(
00283     uint8_t pairingIndex,
00283         EmberRf4cePairingTableEntry *entry
00284 );
00284
00298 EmberStatus emberAfRf4ceGetPairingTableEntry(
00299     uint8_t pairingIndex,
00299         EmberRf4cePairingTableEntry *entry
00300 );
00300
00314 EmberStatus emberAfRf4ceSetApplicationInfo(
00315     EmberRf4ceApplicationInfo *appInfo);
00315
00329 EmberStatus emberAfRf4ceGetApplicationInfo(
00330     EmberRf4ceApplicationInfo *appInfo);
00330
00344 EmberStatus emberAfRf4ceKeyUpdate(uint8_t pairingIndex,
00345     EmberKeyData *key);
00345
00371 EmberStatus emberAfRf4ceSend(uint8_t pairingIndex,
00372     uint8_t profileId,
00373     uint8_t *message,
00374     uint8_t messageLength,
00375     uint8_t *messageTag);
00376
00402 EmberStatus emberAfRf4ceSendVendorSpecific(
00403     uint8_t pairingIndex,
00403         uint8_t profileId,
00404         uint16_t vendorId,
00405         uint8_t *message,
00406         uint8_t messageLength,
00407         uint8_t *messageTag);
00408
00430 EmberStatus emberAfRf4ceSendExtended(uint8_t
00431     pairingIndex,
00431         uint8_t profileId,
00432         uint16_t vendorId,

```

```

00433                     EmberRf4ceTxOption txOptions,
00434                     uint8_t *message,
00435                     uint8_t messageLength,
00436                     uint8_t *messageTag);
00437
00451 EmberStatus emberAfRf4ceGetDefaultTxOptions(
00452     uint8_t pairingIndex,
00453                                         EmberRf4ceTxOption *txOptions);
00468 EmberStatus emberAfRf4ceUnpair(uint8_t pairingIndex);
00469
00483 EmberStatus emberAfRf4ceStop(void);
00484
00498 uint8_t emberAfRf4ceGetMaxPayload(uint8_t pairingIndex
00499
00500                                         EmberRf4ceTxOption txOptions);
00501 #ifdef DOXYGEN_SHOULD_SKIP_THIS
00502
00509     uint8_t emberAfRf4ceDeviceTypeListLength(
00510         EmberRf4ceApplicationCapabilities capabilities);
00510 #else
00511     #define emberAfRf4ceDeviceTypeListLength(capabilities)
00512         (((capabilities) &
00513             EMBER_RF4CE_APP_CAPABILITIES_SUPPORTED_DEVICE_TYPES_MASK) \
00513             >> EMBER_RF4CE_APP_CAPABILITIES_SUPPORTED_DEVICE_TYPES_OFFSET)
00514 #endif
00515
00516 #ifdef DOXYGEN_SHOULD_SKIP_THIS
00517
00524     uint8_t emberAfRf4ceProfileIdListLength(
00525         EmberRf4ceApplicationCapabilities capabilities);
00525 #else
00526     #define emberAfRf4ceProfileIdListLength(capabilities) \
00527         (((capabilities) & EMBER_RF4CE_APP_CAPABILITIES_SUPPORTED_PROFILES_MASK) \
00528             >> EMBER_RF4CE_APP_CAPABILITIES_SUPPORTED_PROFILES_OFFSET)
00529 #endif
00530
00531 #ifdef DOXYGEN_SHOULD_SKIP_THIS
00532
00538     uint16_t emberAfRf4ceVendorId(void);
00539 #else
00540     #define emberAfRf4ceVendorId() (emAfRf4ceVendorInfo.vendorId + 0)
00541 #endif
00542
00551 bool emberAfRf4ceIsDeviceTypeSupported(const
00552     EmberRf4ceApplicationInfo *appInfo,
00553                                         EmberAfRf4ceDeviceType
00554     deviceType);
00555
00554 #ifdef DOXYGEN_SHOULD_SKIP_THIS
00555
00566     bool emberAfRf4ceIsDeviceTypeSupportedLocally
00567     (EmberAfRf4ceDeviceType deviceType);
00568 #else
00568     #define emberAfRf4ceIsDeviceTypeSupportedLocally(deviceType) \
00569         emberAfRf4ceIsDeviceTypeSupported(&emAfRf4ceApplicationInfo, deviceType)
00570 #endif
00571
00576 bool emberAfRf4ceIsProfileSupported(const
00577     EmberRf4ceApplicationInfo *appInfo,
00578                                         EmberAfRf4ceProfileId
00579     profileId);
00579 #ifdef DOXYGEN_SHOULD_SKIP_THIS
00580
00587     bool emberAfRf4ceIsProfileSupportedLocally
00588     (EmberAfRf4ceProfileId profileId);
00589 #else
00589     #define emberAfRf4ceIsProfileSupportedLocally(profileId) \
00590         emberAfRf4ceIsProfileSupported(&emAfRf4ceApplicationInfo, profileId)
00591 #endif
00592
00593 #ifdef DOXYGEN_SHOULD_SKIP_THIS
00594
00600     bool emberAfRf4cePairingTableEntryIsUnused
00601     (const EmberRf4cePairingTableEntry *pairingTableEntry);
00601 #else
00602     #define emberAfRf4cePairingTableEntryIsUnused(pairingTableEntry) \

```

```

00603     (READBITS((pairingTableEntry)->info, \
00604         EMBER_RF4CE_PAIRING_TABLE_ENTRY_INFO_STATUS_MASK) \
00605         == EMBER_RF4CE_PAIRING_TABLE_ENTRY_STATUS_UNUSED) \
00606 #endif \
00607 \
00608 #ifdef DOXYGEN_SHOULD_SKIP_THIS \
00609 \
00615     bool emberAfRf4cePairingTableEntryIsProvisional \
00616     (const EmberRf4cePairingTableEntry *pairingTableEntry); \
00617 #else \
00618     #define emberAfRf4cePairingTableEntryIsProvisional(pairingTableEntry) \
00619     (READBITS((pairingTableEntry)->info, \
00620         EMBER_RF4CE_PAIRING_TABLE_ENTRY_INFO_STATUS_MASK) \
00621         == EMBER_RF4CE_PAIRING_TABLE_ENTRY_STATUS_PROVISIONAL) \
00622 #endif \
00623 \
00623 #ifdef DOXYGEN_SHOULD_SKIP_THIS \
00624 \
00630     bool emberAfRf4cePairingTableEntryIsActive \
00631     (const EmberRf4cePairingTableEntry *pairingTableEntry); \
00632 #else \
00633     #define emberAfRf4cePairingTableEntryIsActive(pairingTableEntry) \
00634     (READBITS((pairingTableEntry)->info, \
00635         EMBER_RF4CE_PAIRING_TABLE_ENTRY_INFO_STATUS_MASK) \
00636         == EMBER_RF4CE_PAIRING_TABLE_ENTRY_STATUS_ACTIVE) \
00637 #endif \
00638 \
00638 #ifdef DOXYGEN_SHOULD_SKIP_THIS \
00639 \
00645     bool emberAfRf4cePairingTableEntryHasLinkKey \
00646     (const EmberRf4cePairingTableEntry *pairingTableEntry); \
00647 #else \
00648     #define emberAfRf4cePairingTableEntryHasLinkKey(pairingTableEntry) \
00649     (READBITS((pairingTableEntry)->info, \
00650         EMBER_RF4CE_PAIRING_TABLE_ENTRY_INFO_HAS_LINK_KEY_BIT) \
00651         == EMBER_RF4CE_PAIRING_TABLE_ENTRY_INFO_HAS_LINK_KEY_BIT) \
00652 #endif \
00653 \
00653 #ifdef DOXYGEN_SHOULD_SKIP_THIS \
00654 \
00662     bool emberAfRf4cePairingTableEntryIsPairingInitiator \
00663     (const EmberRf4cePairingTableEntry *pairingTableEntry); \
00664 #else \
00665     #define emberAfRf4cePairingTableEntryIsPairingInitiator(pairingTableEntry) \
00666     (READBITS((pairingTableEntry)->info, \
00667         EMBER_RF4CE_PAIRING_TABLE_ENTRY_INFO_IS_PAIRING_INITIATOR_BIT) \
00668         == EMBER_RF4CE_PAIRING_TABLE_ENTRY_INFO_IS_PAIRING_INITIATOR_BIT) \
00669 #endif \
00670 \
00670 #ifdef DOXYGEN_SHOULD_SKIP_THIS \
00671 \
00678     bool emberAfRf4cePairingTableEntryHasSecurity \
00679     (const EmberRf4cePairingTableEntry *pairingTableEntry); \
00680 #else \
00681     #define emberAfRf4cePairingTableEntryHasSecurity(pairingTableEntry) \
00682     (READBITS((pairingTableEntry)->capabilities, \
00683         EMBER_RF4CE_NODE_CAPABILITIES_SECURITY_BIT) \
00684         == EMBER_RF4CE_NODE_CAPABILITIES_SECURITY_BIT) \
00685 #endif \
00686 \
00686 #ifdef DOXYGEN_SHOULD_SKIP_THIS \
00687 \
00695     bool emberAfRf4cePairingTableEntryHasChannelNormalization \
00696     (const EmberRf4cePairingTableEntry *pairingTableEntry); \
00697 #else \
00698     #define \
00699     emberAfRf4cePairingTableEntryHasChannelNormalization(pairingTableEntry) \
00700     (READBITS((pairingTableEntry)->capabilities, \
00701         EMBER_RF4CE_NODE_CAPABILITIES_CHANNEL_NORM_BIT) \
00702         == EMBER_RF4CE_NODE_CAPABILITIES_CHANNEL_NORM_BIT) \
00703 #endif \
00704 \
00704 #ifndef DOXYGEN_SHOULD_SKIP_THIS \
00705 extern EmberRf4ceVendorInfo emAfRf4ceVendorInfo; \
00706 extern EmberRf4ceApplicationInfo emAfRf4ceApplicationInfo; \
00707 #endif \
00707 \
00708 // There are a maximum of 128 bytes at the PHY layer, but there is a one-byte

```

```

00709 // length and a two-byte CRC, leaving 125 for the PHY payload. The minimum MAC
00710 // header consists of a two-byte frame control, one-byte sequence number, two-
00711 // byte destination pan id, and at least two bytes each for the source and
00712 // destination addresses. That leaves a maximum of 116 bytes for the MAC
00713 // payload, although inter-pan messages or long source or destination addresses
00714 // will reduce that. The minimum NWK header consists of a one-byte frame
00715 // control, four-byte frame counter, and one-byte profile id. That leaves a
00716 // maximum of 110 bytes for the NWK payload, although encrypted or vendor-
00717 // specific messages will reduce that. Use the ::emberAfRf4ceGetMaxPayload()
00718 // API to retrieve the maximum allowed payload for a certain pairing and
00719 // certain
00720 #define EMBER_AF_RF4CE_MAXIMUM_RF4CE_PAYLOAD_LENGTH 110
00721
00728 #define EMBER_AF_RF4CE_MESSAGE_TAG_MASK 0x7F
00729
00730 #endif // __RF4CE_PROFILE_H__
00731
00732 // @} END addtogroup

```

8.48 rf4ce-zrc11-internal.h File Reference

Macros

- #define NWK_MAX_DISCOVERY_REPETITIONS
- #define NWK_MAX_REPORTED_NODE_DESCRIPTORS
- #define APLC_MAX_CMD_DISC_RX_ON_DURATION_MS
- #define APLC_MAX_KEY_REPEAT_INTERVAL_MS
- #define APLC_MAX_PAIR_INDICATION_WAIT_TIME_MS
- #define APLC_MAX_RESPONSE_WAIT_TIME_MS
- #define APLC_MIN_KEY_EXCHANGE_TRANSFER_COUNT
- #define APLC_MIN_TARGET_BLACKOUT_PERIOD_MS
- #define APLC_DISCOVERY_DURATION_MS
- #define APLC_AUTO_DISCOVERY_RESPONSE_MODE_DURATION_MS
- #define ZRC_FRAME_CONTROL_LENGTH
- #define ZRC_FRAME_CONTROL_COMMAND_CODE_MASK
- #define ZRC_OVERHEAD
- #define COMMAND_CODE_MINIMUM
- #define COMMAND_CODE_MAXIMUM
- #define USER_CONTROL_PRESSED_LENGTH
- #define USER_CONTROL_PRESSED_RC_COMMAND_CODE_OFFSET
- #define USER_CONTROL_PRESSED_RC_COMMAND_PAYLOAD_OFFSET
- #define USER_CONTROL_REPEATED_1_0_LENGTH
- #define USER_CONTROL_REPEATED_1_1_LENGTH
- #define USER_CONTROL_REPEAT_1_1_RC_COMMAND_CODE_OFFSET
- #define USER_CONTROL_REPEAT_1_1_RC_COMMAND_PAYLOAD_OFFSET
- #define USER_CONTROL_RELEASED_1_0_LENGTH
- #define USER_CONTROL_RELEASED_1_1_LENGTH
- #define USER_CONTROL_RELEASED_1_1_RC_COMMAND_CODE_OFFSET
- #define COMMAND_DISCOVERY_REQUEST_LENGTH
- #define COMMANDS_SUPPORTED_LENGTH
- #define COMMAND_DISCOVERY_RESPONSE_LENGTH
- #define COMMAND_DISCOVERY_RESPONSE_COMMANDS_SUPPORTED_OFFSET
- #define EMBER_AF_PLUGIN_RF4CE_ZRC11_MAXIMUM_RC_COMMAND_PAYLOAD_LENGTH
- #define MAXIMUM_USER_CONTROL_X_LENGTH
- #define EMBER_AF_PLUGIN_RF4CE_ZRC11_MAXIMUM_PAYLOAD_LENGTH

Functions

- void `emAfRf4ceZrc11InitOriginator` (void)
- void `emAfRf4ceZrc11InitRecipient` (void)
- EmberStatus `emAfRf4ceZrc11Send` (uint8_t pairingIndex)
- void `emAfRf4ceZrc11RxEnable` (void)
- void `emAfRf4ceZrc11IncomingUserControl` (uint8_t pairingIndex, EmberAfRf4ceZrcCommandCode commandCode, const uint8_t *message, uint8_t messageLength)

Variables

- uint8_t `emAfRf4ceZrc11Buffer` []
- uint8_t `emAfRf4ceZrc11BufferLength`
- EmberEventControl `emberAfPluginRf4ceZrc11PairingEventControl`
- EmberEventControl `emberAfPluginRf4ceZrc11IncomingUserControlEventControl`
- EmberEventControl `emberAfPluginRf4ceZrc11OutgoingUserControlEventControl`
- EmberEventControl `emberAfPluginRf4ceZrc11CommandDiscoveryEventControl`

8.48.1 Macro Definition Documentation

8.48.1.1 #define NWK_MAX_DISCOVERY_REPEATITIONS

Definition at line 4 of file [rf4ce-zrc11-internal.h](#).

8.48.1.2 #define NWK_MAX_REPORTED_NODE_DESCRIPTOROS

Definition at line 7 of file [rf4ce-zrc11-internal.h](#).

8.48.1.3 #define APLC_MAX_CMD_DISC_RX_ON_DURATION_MS

Definition at line 11 of file [rf4ce-zrc11-internal.h](#).

8.48.1.4 #define APLC_MAX_KEY_REPEAT_INTERVAL_MS

Definition at line 15 of file [rf4ce-zrc11-internal.h](#).

8.48.1.5 #define APLC_MAX_PAIR_INDICATION_WAIT_TIME_MS

Definition at line 20 of file [rf4ce-zrc11-internal.h](#).

8.48.1.6 #define APLC_MAX_RESPONSE_WAIT_TIME_MS

Definition at line 24 of file [rf4ce-zrc11-internal.h](#).

8.48.1.7 #define APLC_MIN_KEY_EXCHANGE_TRANSFER_COUNT

Definition at line 28 of file [rf4ce-zrc11-internal.h](#).

8.48.1.8 #define APLC_MIN_TARGET_BLACKOUT_PERIOD_MS

Definition at line 32 of file [rf4ce-zrc11-internal.h](#).

8.48.1.9 #define APLC_DISCOVERY_DURATION_MS

Definition at line 35 of file [rf4ce-zrc11-internal.h](#).

8.48.1.10 #define APLC_AUTO_DISCOVERY_RESPONSE_MODE_DURATION_MS

Definition at line 38 of file [rf4ce-zrc11-internal.h](#).

8.48.1.11 #define ZRC_FRAME_CONTROL_LENGTH

Definition at line 43 of file [rf4ce-zrc11-internal.h](#).

8.48.1.12 #define ZRC_FRAME_CONTROL_COMMAND_CODE_MASK

Definition at line 44 of file [rf4ce-zrc11-internal.h](#).

8.48.1.13 #define ZRC_OVERHEAD

Definition at line 45 of file [rf4ce-zrc11-internal.h](#).

8.48.1.14 #define COMMAND_CODE_MINIMUM

Definition at line 47 of file [rf4ce-zrc11-internal.h](#).

8.48.1.15 #define COMMAND_CODE_MAXIMUM

Definition at line 48 of file [rf4ce-zrc11-internal.h](#).

8.48.1.16 #define USER_CONTROL_PRESSED_LENGTH

Definition at line 53 of file [rf4ce-zrc11-internal.h](#).

8.48.1.17 #define USER_CONTROL_PRESSED_RC_COMMAND_CODE_OFFSET

Definition at line 54 of file [rf4ce-zrc11-internal.h](#).

8.48.1.18 #define USER_CONTROL_PRESSED_RC_COMMAND_PAYLOAD_OFFSET

Definition at line 55 of file [rf4ce-zrc11-internal.h](#).

8.48.1.19 #define USER_CONTROL_REPEAT_1_0_LENGTH

Definition at line 60 of file [rf4ce-zrc11-internal.h](#).

8.48.1.20 #define USER_CONTROL_REPEAT_1_1_LENGTH

Definition at line 61 of file [rf4ce-zrc11-internal.h](#).

8.48.1.21 #define USER_CONTROL_REPEAT_1_1_RC_COMMAND_CODE_OFFSET

Definition at line 62 of file [rf4ce-zrc11-internal.h](#).

8.48.1.22 #define USER_CONTROL_REPEAT_1_1_RC_COMMAND_PAYLOAD_OFFSET

Definition at line 63 of file [rf4ce-zrc11-internal.h](#).

8.48.1.23 #define USER_CONTROL_RELEASED_1_0_LENGTH

Definition at line 67 of file [rf4ce-zrc11-internal.h](#).

8.48.1.24 #define USER_CONTROL_RELEASED_1_1_LENGTH

Definition at line 68 of file [rf4ce-zrc11-internal.h](#).

8.48.1.25 #define USER_CONTROL_RELEASED_1_1_RC_COMMAND_CODE_OFFSET

Definition at line 69 of file [rf4ce-zrc11-internal.h](#).

8.48.1.26 #define COMMAND_DISCOVERY_REQUEST_LENGTH

Definition at line 73 of file [rf4ce-zrc11-internal.h](#).

8.48.1.27 #define COMMANDS_SUPPORTED_LENGTH

Definition at line 78 of file [rf4ce-zrc11-internal.h](#).

8.48.1.28 #define COMMAND_DISCOVERY_RESPONSE_LENGTH

Definition at line 79 of file [rf4ce-zrc11-internal.h](#).

8.48.1.29 #define COMMAND_DISCOVERY_RESPONSE_COMMANDS_SUPPORTED_OFFSET

Definition at line 80 of file [rf4ce-zrc11-internal.h](#).

8.48.1.30 #define EMBER_AF_PLUGIN_RF4CE_ZRC11_MAXIMUM_RC_COMMAND_PAYLOAD_LENGTH

Definition at line 87 of file [rf4ce-zrc11-internal.h](#).

8.48.1.31 #define MAXIMUM_USER_CONTROL_X_LENGTH

Definition at line 89 of file [rf4ce-zrc11-internal.h](#).

8.48.1.32 #define EMBER_AF_PLUGIN_RF4CE_ZRC11_MAXIMUM_PAYLOAD_LENGTH

Definition at line 97 of file [rf4ce-zrc11-internal.h](#).

8.48.2 Function Documentation

8.48.2.1 void emAfRf4ceZrc11InitOriginator (void)

8.48.2.2 void emAfRf4ceZrc11InitRecipient (void)

8.48.2.3 EmberStatus emAfRf4ceZrc11Send (uint8_t *pairingIndex*)

8.48.2.4 void emAfRf4ceZrc11RxEnable (void)

8.48.2.5 void emAfRf4ceZrc11IncomingUserControl (uint8_t *pairingIndex*, EmberAfRf4ceZrc-
CommandCode *commandCode*, const uint8_t * *message*, uint8_t *messageLength*
)

8.48.3 Variable Documentation

8.48.3.1 uint8_t emAfRf4ceZrc11Buffer[]

8.48.3.2 uint8_t emAfRf4ceZrc11BufferLength

8.48.3.3 EmberEventControl emberAfPluginRf4ceZrc11PairingEventControl

8.48.3.4 EmberEventControl emberAfPluginRf4ceZrc11IncomingUserControlEventControl

8.48.3.5 EmberEventControl emberAfPluginRf4ceZrc11OutgoingUserControlEventControl

8.48.3.6 EmberEventControl emberAfPluginRf4ceZrc11CommandDiscoveryEventControl

8.49 rf4ce-zrc11-internal.h

```

00001 // Copyright 2014 Silicon Laboratories, Inc.
00002
00003 // The maximum number of repetitions performed during discovery.
00004 #define NWK_MAX_DISCOVERY_REPEATITIONS 30
00005
00006 // The maximum number of node descriptors reported during discovery.
00007 #define NWK_MAX_REPORTED_NODE_DESCRIPTOROS 1
00008
00009 // The maximum duration that the receiver is enabled on a controller after
00010 // pairing to receive any command discovery request command frames.
00011 #define APLC_MAX_CMD_DISC_RX_ON_DURATION_MS 200
00012

```

```

00013 // The maximum time between consecutive user control repeated command frame
00014 // transmissions.
00015 #define APLC_MAX_KEY_REPEAT_INTERVAL_MS 100
00016
00017 // The maximum amount of time a device waits after receiving a successful
00018 // NLME-AUTODISCOVERY.confirm primitive for a pair indication to arrive from
00019 // the pairing initiator.
00020 #define APLC_MAX_PAIR_INDICATION_WAIT_TIME_MS MILLISECOND_TICKS_PER_SECOND
00021
00022 // The maximum time a device shall wait for a response command frame following
00023 // a request command frame.
00024 #define APLC_MAX_RESPONSE_WAIT_TIME_MS 200
00025
00026 // The minimum value of the KeyExTransferCount parameter passed to the pair
00027 // request primitive during the push button pairing procedure.
00028 #define APLC_MIN_KEY_EXCHANGE_TRANSFER_COUNT 3
00029
00030 // The minimum amount of time a device must wait after a successful pairing
00031 // attempt with a target before attempting command discovery.
00032 #define APLC_MIN_TARGET_BLACKOUT_PERIOD_MS 500
00033
00034 // The amount of time a device must perform discovery.
00035 #define APLC_DISCOVERY_DURATION_MS 100
00036
00037 // The amount of time a device must wait in automatic discovery response mode.
00038 #define APLC_AUTO_DISCOVERY_RESPONSE_MODE_DURATION_MS \
00039     (30 * MILLISECOND_TICKS_PER_SECOND)
00040
00041 // ZRC header
00042 // - Frame control (1 byte)
00043 #define ZRC_FRAME_CONTROL_LENGTH 1
00044 #define ZRC_FRAME_CONTROL_COMMAND_CODE_MASK 0x1F
00045 #define ZRC_OVERHEAD (ZRC_FRAME_CONTROL_LENGTH)
00046
00047 #define COMMAND_CODE_MINIMUM EMBER_AF_RF4CE_ZRC_COMMAND_USER_CONTROL_PRESSED
00048 #define COMMAND_CODE_MAXIMUM
    EMBER_AF_RF4CE_ZRC_COMMAND_COMMAND_DISCOVERY_RESPONSE
00049
00050 // User Control Pressed
00051 // - RC command code (1 byte)
00052 // - RC command payload (n bytes)
00053 #define USER_CONTROL_PRESSED_LENGTH (ZRC_OVERHEAD + 1)
00054 #define USER_CONTROL_PRESSED_RC_COMMAND_CODE_OFFSET (ZRC_OVERHEAD)
00055 #define USER_CONTROL_PRESSED_RC_COMMAND_PAYLOAD_OFFSET (ZRC_OVERHEAD + 1)
00056
00057 // User Control Repeated
00058 // - RC command code (1 byte, 1.1 only)
00059 // - RC command payload (n bytes, 1.1 only)
00060 #define USER_CONTROL_REPEATED_1_0_LENGTH (ZRC_OVERHEAD)
00061 #define USER_CONTROL_REPEATED_1_1_LENGTH (ZRC_OVERHEAD + 1)
00062 #define USER_CONTROL_REPEATED_1_1_RC_COMMAND_CODE_OFFSET (ZRC_OVERHEAD)
00063 #define USER_CONTROL_REPEATED_1_1_RC_COMMAND_PAYLOAD_OFFSET (ZRC_OVERHEAD + 1)
00064
00065 // User Control Released
00066 // - RC command code (1 byte, 1.1 only)
00067 #define USER_CONTROL_RELEASED_1_0_LENGTH (ZRC_OVERHEAD)
00068 #define USER_CONTROL_RELEASED_1_1_LENGTH (ZRC_OVERHEAD + 1)
00069 #define USER_CONTROL_RELEASED_1_1_RC_COMMAND_CODE_OFFSET (ZRC_OVERHEAD)
00070
00071 // Command Discovery Request
00072 // - Reserved (1 byte)
00073 #define COMMAND_DISCOVERY_REQUEST_LENGTH (ZRC_OVERHEAD + 1)
00074
00075 // Command Discovery Response
00076 // - Reserved (1 byte)
00077 // - Commands supported (32 bytes)
00078 #define COMMANDS_SUPPORTED_LENGTH 32
00079 #define COMMAND_DISCOVERY_RESPONSE_LENGTH (ZRC_OVERHEAD + 1 +
    COMMANDS_SUPPORTED_LENGTH)
00080 #define COMMAND_DISCOVERY_RESPONSE_COMMANDS_SUPPORTED_OFFSET (ZRC_OVERHEAD + 1)
00081
00082 // The User Control Pressed and User Control Repeated commands theoretically
00083 // take an unbounded additional payload, but the longest additional operand in
00084 // HDMI 1.3a is just four bytes. Still, just in case, leave an opening for the
00085 // user to override the buffer size.
00086 #ifndef EMBER_AF_PLUGIN_RF4CE_ZRC11_MAXIMUM_RC_COMMAND_PAYLOAD_LENGTH
00087     #define EMBER_AF_PLUGIN_RF4CE_ZRC11_MAXIMUM_RC_COMMAND_PAYLOAD_LENGTH 4
00088 #endif
00089 #define MAXIMUM_USER_CONTROL_X_LENGTH \
00090     (USER_CONTROL_PRESSED_LENGTH) \

```

```

00091     + EMBER_AF_PLUGIN_RF4CE_ZRC11_MAXIMUM_RC_COMMAND_PAYLOAD_LENGTH)
00092
00093 // Assuming the standard operands are used, the Discovery Response command is
00094 // the command with the longest payload in the ZRC profile.
00095 #ifndef EMBER_AF_PLUGIN_RF4CE_ZRC11_MAXIMUM_PAYLOAD_LENGTH
00096     #if MAXIMUM_USER_CONTROL_X_LENGTH < COMMAND_DISCOVERY_RESPONSE_LENGTH
00097         #define EMBER_AF_PLUGIN_RF4CE_ZRC11_MAXIMUM_PAYLOAD_LENGTH
00098             COMMAND_DISCOVERY_RESPONSE_LENGTH
00099     #else
00100         #define EMBER_AF_PLUGIN_RF4CE_ZRC11_MAXIMUM_PAYLOAD_LENGTH
00101             MAXIMUM_USER_CONTROL_X_LENGTH
00102     #endif
00103 #endif
00104
00105
00106 extern uint8_t emAfRf4ceZrc11Buffer[];
00107 extern uint8_t emAfRf4ceZrc11BufferLength;
00108 EmberStatus emAfRf4ceZrc11Send(uint8_t pairingIndex);
00109
00110 extern EmberEventControl emberAfPluginRf4ceZrc11PairingEventControl
00111 ;
00112 extern EmberEventControl emberAfPluginRf4ceZrc11IncomingUserControlEventControl
00113 extern EmberEventControl emberAfPluginRf4ceZrc11OutgoingUserControlEventControl
00114 ;
00115 void emAfRf4ceZrc11RxEnable(void);
00116
00117 void emAfRf4ceZrc11IncomingUserControl(uint8_t
00118     pairingIndex,
00119     commandCode,
00120     const uint8_t *message,
00121     uint8_t messageLength);

```

8.50 rf4ce-zrc11-types.h File Reference

Data Structures

- struct [EmberAfRf4ceZrcUserControlRecord](#)
This data structure contains the ZRC 1.x user control record.
- struct [EmberAfRf4ceZrcCommandsSupported](#)
This data structure contains the ZRC 1.x command discovery data.

Macros

- #define [EMBER_AF_RF4CE_ZRC_COMMANDS_SUPPORTED_SIZE](#)

Enumerations

- enum EmberAfRf4ceZrcCommandCode {
EMBER_AF_RF4CE_ZRC_COMMAND_USER_CONTROL_PRESSED, EMBER_AF_RF4CE_ZRC_COMMAND_USER_CONTROL_REPEAT, EMBER_AF_RF4CE_ZRC_COMMAND_USER_CONTROL_RELEASED, EMBER_AF_RF4CE_ZRC_COMMAND_COMMAND_DISCOVERY_REQUEST,
EMBER_AF_RF4CE_ZRC_COMMAND_COMMAND_DISCOVERY_RESPONSE, EMBER_AF_RF4CE_ZRC_COMMAND_USER_CONTROL_PRESSED, EMBER_AF_RF4CE_ZRC_COMMAND_USER_CONTROL_REPEAT, EMBER_AF_RF4CE_ZRC_COMMAND_USER_CONTROL_RELEASED,
EMBER_AF_RF4CE_ZRC_COMMAND_COMMAND_DISCOVERY_REQUEST, EMBER_AF_RF4CE_ZRC_COMMAND_COMMAND_DISCOVERY_RESPONSE, EMBER_AF_RF4CE_ZRC_COMMAND_ACTIONS }

- enum EmberAfRf4ceUserControlCode {
 EMBER_AF_RF4CE_USER_CONTROL_CODE_SELECT, EMBER_AF_RF4CE_USER_CONTROL_CODE_UP, EMBER_AF_RF4CE_USER_CONTROL_CODE_DOWN, EMBER_AF_RF4CE_USER_CONTROL_CODE_LEFT,
 EMBER_AF_RF4CE_USER_CONTROL_CODE_RIGHT, EMBER_AF_RF4CE_USER_CONTROL_CODE_UP, EMBER_AF_RF4CE_USER_CONTROL_CODE_DOWN, EMBER_AF_RF4CE_USER_CONTROL_CODE_LEFT_UP,
 EMBER_AF_RF4CE_USER_CONTROL_CODE_UP, EMBER_AF_RF4CE_USER_CONTROL_CODE_DOWN, EMBER_AF_RF4CE_USER_CONTROL_CODE_ROOT_MENU, EMBER_AF_RF4CE_USER_CONTROL_CODE_SETUP_MENU, EMBER_AF_RF4CE_USER_CONTROL_CODE_CONTENTS_MENU,
 EMBER_AF_RF4CE_USER_CONTROL_CODE_FAVORITE_MENU, EMBER_AF_RF4CE_USER_CONTROL_CODE_EXIT, EMBER_AF_RF4CE_USER_CONTROL_CODE_0, EMBER_AF_RF4CE_USER_CONTROL_CODE_1,
 EMBER_AF_RF4CE_USER_CONTROL_CODE_2, EMBER_AF_RF4CE_USER_CONTROL_CODE_3, EMBER_AF_RF4CE_USER_CONTROL_CODE_4, EMBER_AF_RF4CE_USER_CONTROL_CODE_5,
 EMBER_AF_RF4CE_USER_CONTROL_CODE_6, EMBER_AF_RF4CE_USER_CONTROL_CODE_7, EMBER_AF_RF4CE_USER_CONTROL_CODE_8, EMBER_AF_RF4CE_USER_CONTROL_CODE_9,
 EMBER_AF_RF4CE_USER_CONTROL_CODE_DOT, EMBER_AF_RF4CE_USER_CONTROL_CODE_ENTER, EMBER_AF_RF4CE_USER_CONTROL_CODE_CLEAR, EMBER_AF_RF4CE_USER_CONTROL_CODE_NEXT_FAVORITE,
 EMBER_AF_RF4CE_USER_CONTROL_CODE_CHANNEL_UP, EMBER_AF_RF4CE_USER_CONTROL_CODE_CHANNEL_DOWN, EMBER_AF_RF4CE_USER_CONTROL_CODE_PREVIOUS_CHANNEL, EMBER_AF_RF4CE_USER_CONTROL_CODE_SOUND_SELECT,
 EMBER_AF_RF4CE_USER_CONTROL_CODE_INPUT_SELECT, EMBER_AF_RF4CE_USER_CONTROL_CODE_DISPLAY_INFORMATION, EMBER_AF_RF4CE_USER_CONTROL_CODE_HELP, EMBER_AF_RF4CE_USER_CONTROL_CODE_PAGE_UP,
 EMBER_AF_RF4CE_USER_CONTROL_CODE_PAGE_DOWN, EMBER_AF_RF4CE_USER_CONTROL_CODE_POWER, EMBER_AF_RF4CE_USER_CONTROL_CODE_VOLUME_UP, EMBER_AF_RF4CE_USER_CONTROL_CODE_VOLUME_DOWN,
 EMBER_AF_RF4CE_USER_CONTROL_CODE_MUTE, EMBER_AF_RF4CE_USER_CONTROL_CODE_PLAY, EMBER_AF_RF4CE_USER_CONTROL_CODE_STOP, EMBER_AF_RF4CE_USER_CONTROL_CODE_PAUSE,
 EMBER_AF_RF4CE_USER_CONTROL_CODE_RECORD, EMBER_AF_RF4CE_USER_CONTROL_CODE_REWIND, EMBER_AF_RF4CE_USER_CONTROL_CODE_FAST_FORWARD, EMBER_AF_RF4CE_USER_CONTROL_CODE_EJECT,
 EMBER_AF_RF4CE_USER_CONTROL_CODE_FORWARD, EMBER_AF_RF4CE_USER_CONTROL_CODE_BACKWARD, EMBER_AF_RF4CE_USER_CONTROL_CODE_STOP_RECORD, EMBER_AF_RF4CE_USER_CONTROL_CODE_PAUSE_RECORD,
 EMBER_AF_RF4CE_USER_CONTROL_CODE_ANGLE, EMBER_AF_RF4CE_USER_CONTROL_CODE_SUB_PICTURE, EMBER_AF_RF4CE_USER_CONTROL_CODE_VIDEO_ON_DEMAND, EMBER_AF_RF4CE_USER_CONTROL_CODE_ELECTRONIC_PROGRAM_GUIDE,
 EMBER_AF_RF4CE_USER_CONTROL_CODE_TIMER_PROGRAMMING, EMBER_AF_RF4CE_USER_CONTROL_CODE_INITIAL_CONFIGURATION, EMBER_AF_RF4CE_USER_CONTROL_CODE_PLAY_FUNCTION, EMBER_AF_RF4CE_USER_CONTROL_CODE_PAUSE_PLAY_FUNCTION,
 EMBER_AF_RF4CE_USER_CONTROL_CODE_RECORD_FUNCTION, EMBER_AF_RF4CE_USER_CONTROL_CODE_PAUSE_RECORD_FUNCTION, EMBER_AF_RF4CE_USER_CONTROL_CODE_STOP_FUNCTION, EMBER_AF_RF4CE_USER_CONTROL_CODE_MUTE_FUNCTION,
 EMBER_AF_RF4CE_USER_CONTROL_CODE_RESTORE_VOLUME_FUNCTION, EMBER_AF_RF4CE_USER_CONTROL_CODE_TUNE_FUNCTION, EMBER_AF_RF4CE_USER_CONTROL_CODE_SELECT_MEDIA_FUNCTION, EMBER_AF_RF4CE_USER_CONTROL_CODE_SELECT_AUDIO_INPUT_FUNCTION, EMBER_AF_RF4CE_USER_CONTROL_CODE_POWER_TOGGLE_FUNCTION
 }

8.50.1 Macro Definition Documentation

8.50.1.1 `#define EMBER_AF_RF4CE_ZRC_COMMANDS_SUPPORTED_SIZE`

Size of the ZRC 1.x command discovery data in bytes (32).

Definition at line 128 of file [rf4ce-zrc11-types.h](#).

8.50.2 Enumeration Type Documentation

8.50.2.1 `enum EmberAfRf4ceZrcCommandCode`

RF4CE ZRC 1.1 command codes.

Enumerator:

```
EMBER_AF_RF4CE_ZRC_COMMAND_USER_CONTROL_PRESSED
EMBER_AF_RF4CE_ZRC_COMMAND_USER_CONTROL_REPEATED
EMBER_AF_RF4CE_ZRC_COMMAND_USER_CONTROL_RELEASED
EMBER_AF_RF4CE_ZRC_COMMAND_COMMAND_DISCOVERY_REQUEST
EMBER_AF_RF4CE_ZRC_COMMAND_COMMAND_DISCOVERY_RESPONSE
EMBER_AF_RF4CE_ZRC_COMMAND_USER_CONTROL_PRESSED
EMBER_AF_RF4CE_ZRC_COMMAND_USER_CONTROL_REPEATED
EMBER_AF_RF4CE_ZRC_COMMAND_USER_CONTROL_RELEASED
EMBER_AF_RF4CE_ZRC_COMMAND_COMMAND_DISCOVERY_REQUEST
EMBER_AF_RF4CE_ZRC_COMMAND_COMMAND_DISCOVERY_RESPONSE
EMBER_AF_RF4CE_ZRC_COMMAND_ACTIONS
```

Definition at line 10 of file [rf4ce-zrc11-types.h](#).

8.50.2.2 `enum EmberAfRf4ceUserControlCode`

RF4CE user control codes from HDMI 1.3a CEC operand [UI Command].

Enumerator:

```
EMBER_AF_RF4CE_USER_CONTROL_CODE_SELECT
EMBER_AF_RF4CE_USER_CONTROL_CODE_UP
EMBER_AF_RF4CE_USER_CONTROL_CODE_DOWN
EMBER_AF_RF4CE_USER_CONTROL_CODE_LEFT
EMBER_AF_RF4CE_USER_CONTROL_CODE_RIGHT
EMBER_AF_RF4CE_USER_CONTROL_CODE_RIGHT_UP
EMBER_AF_RF4CE_USER_CONTROL_CODE_RIGHT_DOWN
EMBER_AF_RF4CE_USER_CONTROL_CODE_LEFT_UP
EMBER_AF_RF4CE_USER_CONTROL_CODE_LEFT_DOWN
EMBER_AF_RF4CE_USER_CONTROL_CODE_ROOT_MENU
EMBER_AF_RF4CE_USER_CONTROL_CODE_SETUP_MENU
```

EMBER_AF_RF4CE_USER_CONTROL_CODE_CONTENTS_MENU
EMBER_AF_RF4CE_USER_CONTROL_CODE_FAVORITE_MENU
EMBER_AF_RF4CE_USER_CONTROL_CODE_EXIT
EMBER_AF_RF4CE_USER_CONTROL_CODE_0
EMBER_AF_RF4CE_USER_CONTROL_CODE_1
EMBER_AF_RF4CE_USER_CONTROL_CODE_2
EMBER_AF_RF4CE_USER_CONTROL_CODE_3
EMBER_AF_RF4CE_USER_CONTROL_CODE_4
EMBER_AF_RF4CE_USER_CONTROL_CODE_5
EMBER_AF_RF4CE_USER_CONTROL_CODE_6
EMBER_AF_RF4CE_USER_CONTROL_CODE_7
EMBER_AF_RF4CE_USER_CONTROL_CODE_8
EMBER_AF_RF4CE_USER_CONTROL_CODE_9
EMBER_AF_RF4CE_USER_CONTROL_CODE_DOT
EMBER_AF_RF4CE_USER_CONTROL_CODE_ENTER
EMBER_AF_RF4CE_USER_CONTROL_CODE_CLEAR
EMBER_AF_RF4CE_USER_CONTROL_CODE_NEXT_FAVORITE
EMBER_AF_RF4CE_USER_CONTROL_CODE_CHANNEL_UP
EMBER_AF_RF4CE_USER_CONTROL_CODE_CHANNEL_DOWN
EMBER_AF_RF4CE_USER_CONTROL_CODE_PREVIOUS_CHANNEL
EMBER_AF_RF4CE_USER_CONTROL_CODE_SOUND_SELECT
EMBER_AF_RF4CE_USER_CONTROL_CODE_INPUT_SELECT
EMBER_AF_RF4CE_USER_CONTROL_CODE_DISPLAY_INFORMATION
EMBER_AF_RF4CE_USER_CONTROL_CODE_HELP
EMBER_AF_RF4CE_USER_CONTROL_CODE_PAGE_UP
EMBER_AF_RF4CE_USER_CONTROL_CODE_PAGE_DOWN
EMBER_AF_RF4CE_USER_CONTROL_CODE_POWER
EMBER_AF_RF4CE_USER_CONTROL_CODE_VOLUME_UP
EMBER_AF_RF4CE_USER_CONTROL_CODE_VOLUME_DOWN
EMBER_AF_RF4CE_USER_CONTROL_CODE_MUTE
EMBER_AF_RF4CE_USER_CONTROL_CODE_PLAY
EMBER_AF_RF4CE_USER_CONTROL_CODE_STOP
EMBER_AF_RF4CE_USER_CONTROL_CODE_PAUSE
EMBER_AF_RF4CE_USER_CONTROL_CODE_RECORD
EMBER_AF_RF4CE_USER_CONTROL_CODE_REWIND
EMBER_AF_RF4CE_USER_CONTROL_CODE_FAST_FORWARD
EMBER_AF_RF4CE_USER_CONTROL_CODE_EJECT
EMBER_AF_RF4CE_USER_CONTROL_CODE_FORWARD
EMBER_AF_RF4CE_USER_CONTROL_CODE_BACKWARD
EMBER_AF_RF4CE_USER_CONTROL_CODE_STOP_RECORD
EMBER_AF_RF4CE_USER_CONTROL_CODE_PAUSE_RECORD

```

EMBER_AF_RF4CE_USER_CONTROL_CODE_ANGLE
EMBER_AF_RF4CE_USER_CONTROL_CODE_SUB_PICTURE
EMBER_AF_RF4CE_USER_CONTROL_CODE_VIDEO_ON_DEMAND
EMBER_AF_RF4CE_USER_CONTROL_CODE ELECTRONIC_PROGRAM_GUIDE
EMBER_AF_RF4CE_USER_CONTROL_CODE_TIMER_PROGRAMMING
EMBER_AF_RF4CE_USER_CONTROL_CODE_INITIAL_CONFIGURATION
EMBER_AF_RF4CE_USER_CONTROL_CODE_PLAY_FUNCTION
EMBER_AF_RF4CE_USER_CONTROL_CODE_PAUSE_PLAY_FUNCTION
EMBER_AF_RF4CE_USER_CONTROL_CODE_RECORD_FUNCTION
EMBER_AF_RF4CE_USER_CONTROL_CODE_PAUSE_RECORD_FUNCTION
EMBER_AF_RF4CE_USER_CONTROL_CODE_STOP_FUNCTION
EMBER_AF_RF4CE_USER_CONTROL_CODE_MUTE_FUNCTION
EMBER_AF_RF4CE_USER_CONTROL_CODE_RESTORE_VOLUME_FUNCTION
EMBER_AF_RF4CE_USER_CONTROL_CODE_TUNE_FUNCTION
EMBER_AF_RF4CE_USER_CONTROL_CODE_SELECT_MEDIA_FUNCTION
EMBER_AF_RF4CE_USER_CONTROL_CODE_SELECT_A_V_INPUT_FUNCTION
EMBER_AF_RF4CE_USER_CONTROL_CODE_SELECT_AUDIO_INPUT_FUNCTION
EMBER_AF_RF4CE_USER_CONTROL_CODE_POWER_TOGGLE_FUNCTION
EMBER_AF_RF4CE_USER_CONTROL_CODE_POWER_OFF_FUNCTION
EMBER_AF_RF4CE_USER_CONTROL_CODE_POWER_ON_FUNCTION
EMBER_AF_RF4CE_USER_CONTROL_CODE_F1_BLUE
EMBER_AF_RF4CE_USER_CONTROL_CODE_F2_RED
EMBER_AF_RF4CE_USER_CONTROL_CODE_F3_GREEN
EMBER_AF_RF4CE_USER_CONTROL_CODE_F4_YELLOW
EMBER_AF_RF4CE_USER_CONTROL_CODE_F5
EMBER_AF_RF4CE_USER_CONTROL_CODE_DATA

```

Definition at line 27 of file [rf4ce-zrc11-types.h](#).

8.51 rf4ce-zrc11-types.h

```

00001 // Copyright 2014 Silicon Laboratories, Inc.
00002
00003 #ifndef __RF4CE_ZRC11_TYPES_H__
00004 #define __RF4CE_ZRC11_TYPES_H__
00005
00009 #ifdef DOXYGEN_SHOULD_SKIP_THIS
00010 enum EmberAfRf4ceZrcCommandCode
00011 {
00012     typedef uint8_t EmberAfRf4ceZrcCommandCode;
00013     enum
00014     {
00015         EMBER_AF_RF4CE_ZRC_COMMAND_USER_CONTROL_PRESSED
00016             = 0x01,
00017         EMBER_AF_RF4CE_ZRC_COMMAND_USER_CONTROL_REPEAT
00018             = 0x02,
00019         EMBER_AF_RF4CE_ZRC_COMMAND_USER_CONTROL_RELEASED
00020             = 0x03,
00021         EMBER_AF_RF4CE_ZRC_COMMAND_DISCOVERY_REQUEST
00022             = 0x04,

```

```

00020     EMBER_AF_RF4CE_ZRC_COMMAND_COMMAND_DISCOVERY_RESPONSE
00021     = 0x05,
00022 };
00023
00024 #ifdef DOXYGEN_SHOULD_SKIP_THIS
00025 enum EmberAfRf4ceUserControlCode
00026 #else
00027 typedef uint8_t EmberAfRf4ceUserControlCode;
00028 enum
00029 #endif
00030 {
00031
00032     EMBER_AF_RF4CE_USER_CONTROL_CODE_SELECT
00033     = 0x00,
00034     EMBER_AF_RF4CE_USER_CONTROL_CODE_UP
00035     = 0x01,
00036     EMBER_AF_RF4CE_USER_CONTROL_CODE_DOWN
00037     = 0x02,
00038     EMBER_AF_RF4CE_USER_CONTROL_CODE_LEFT
00039     = 0x03,
00040     EMBER_AF_RF4CE_USER_CONTROL_CODE_RIGHT
00041     = 0x04,
00042     EMBER_AF_RF4CE_USER_CONTROL_CODE_RIGHT_UP
00043     = 0x05,
00044     EMBER_AF_RF4CE_USER_CONTROL_CODE_RIGHT_DOWN
00045     = 0x06,
00046     EMBER_AF_RF4CE_USER_CONTROL_CODE_LEFT_UP
00047     = 0x07,
00048     EMBER_AF_RF4CE_USER_CONTROL_CODE_LEFT_DOWN
00049     = 0x08,
00050     EMBER_AF_RF4CE_USER_CONTROL_CODE_ROOT_MENU
00051     = 0x09,
00052     EMBER_AF_RF4CE_USER_CONTROL_CODE_SETUP_MENU
00053     = 0x0A,
00054     EMBER_AF_RF4CE_USER_CONTROL_CODE_CONTENTS_MENU
00055     = 0x0B,
00056     EMBER_AF_RF4CE_USER_CONTROL_CODE_FAVORITE_MENU
00057     = 0x0C,
00058     EMBER_AF_RF4CE_USER_CONTROL_CODE_EXIT
00059     = 0x0D,
00060     EMBER_AF_RF4CE_USER_CONTROL_CODE_0
00061     = 0x20,
00062     EMBER_AF_RF4CE_USER_CONTROL_CODE_1
00063     = 0x21,
00064     EMBER_AF_RF4CE_USER_CONTROL_CODE_2
00065     = 0x22,
00066     EMBER_AF_RF4CE_USER_CONTROL_CODE_3
00067     = 0x23,
00068     EMBER_AF_RF4CE_USER_CONTROL_CODE_4
00069     = 0x24,
00070     EMBER_AF_RF4CE_USER_CONTROL_CODE_5
00071     = 0x25,
00072     EMBER_AF_RF4CE_USER_CONTROL_CODE_6
00073     = 0x26,
00074     EMBER_AF_RF4CE_USER_CONTROL_CODE_7
00075     = 0x27,
00076     EMBER_AF_RF4CE_USER_CONTROL_CODE_8
00077     = 0x28,
00078     EMBER_AF_RF4CE_USER_CONTROL_CODE_9
00079     = 0x29,
00080     EMBER_AF_RF4CE_USER_CONTROL_CODE_DOT
00081     = 0x2A,
00082     EMBER_AF_RF4CE_USER_CONTROL_CODE_ENTER
00083     = 0x2B,
00084     EMBER_AF_RF4CE_USER_CONTROL_CODE_CLEAR
00085     = 0x2C,
00086     EMBER_AF_RF4CE_USER_CONTROL_CODE_NEXT_FAVORITE
00087     = 0x2F,
00088     EMBER_AF_RF4CE_USER_CONTROL_CODE_CHANNEL_UP
00089     = 0x30,
00090     EMBER_AF_RF4CE_USER_CONTROL_CODE_CHANNEL_DOWN
00091     = 0x31,
00092     EMBER_AF_RF4CE_USER_CONTROL_CODE_PREVIOUS_CHANNEL
00093     = 0x32,
00094     EMBER_AF_RF4CE_USER_CONTROL_CODE_SOUND_SELECT
00095     = 0x33,
00096     EMBER_AF_RF4CE_USER_CONTROL_CODE_INPUT_SELECT
00097     = 0x34,
00098     EMBER_AF_RF4CE_USER_CONTROL_CODE_DISPLAY_INFORMATION
00099     = 0x35,
00100    EMBER_AF_RF4CE_USER_CONTROL_CODE_HELP

```

```

00068     = 0x36,
EMBER_AF_RF4CE_USER_CONTROL_CODE_PAGE_UP
00069     = 0x37,
EMBER_AF_RF4CE_USER_CONTROL_CODE_PAGE_DOWN
00070     = 0x38,
EMBER_AF_RF4CE_USER_CONTROL_CODE_POWER
00071     = 0x40,
EMBER_AF_RF4CE_USER_CONTROL_CODE_VOLUME_UP
00072     = 0x41,
EMBER_AF_RF4CE_USER_CONTROL_CODE_VOLUME_DOWN
00073     = 0x42,
EMBER_AF_RF4CE_USER_CONTROL_CODE_MUTE
00074     = 0x43,
EMBER_AF_RF4CE_USER_CONTROL_CODE_PLAY
00075     = 0x44,
EMBER_AF_RF4CE_USER_CONTROL_CODE_STOP
00076     = 0x45,
EMBER_AF_RF4CE_USER_CONTROL_CODE_PAUSE
00077     = 0x46,
EMBER_AF_RF4CE_USER_CONTROL_CODE_RECORD
00078     = 0x47,
EMBER_AF_RF4CE_USER_CONTROL_CODE_REWIND
00079     = 0x48,
EMBER_AF_RF4CE_USER_CONTROL_CODE_FAST_FORWARD
00080     = 0x49,
EMBER_AF_RF4CE_USER_CONTROL_CODE_EJECT
00081     = 0x4A,
EMBER_AF_RF4CE_USER_CONTROL_CODE_FORWARD
00082     = 0x4B,
EMBER_AF_RF4CE_USER_CONTROL_CODE_BACKWARD
00083     = 0x4C,
EMBER_AF_RF4CE_USER_CONTROL_CODE_STOP_RECORD
00084     = 0x4D,
EMBER_AF_RF4CE_USER_CONTROL_CODE_PAUSE_RECORD
00085     = 0x4E,
EMBER_AF_RF4CE_USER_CONTROL_CODE_ANGLE
00086     = 0x50,
EMBER_AF_RF4CE_USER_CONTROL_CODE_SUB_PICTURE
00087     = 0x51,
EMBER_AF_RF4CE_USER_CONTROL_CODE_VIDEO_ON_DEMAND
00088     = 0x52,
EMBER_AF_RF4CE_USER_CONTROL_CODE ELECTRONIC_PROGRAM_GUIDE
00089     = 0x53,
EMBER_AF_RF4CE_USER_CONTROL_CODE_TIMER_PROGRAMMING
00090     = 0x54,
EMBER_AF_RF4CE_USER_CONTROL_CODE_INITIAL_CONFIGURATION
00091     = 0x55,
EMBER_AF_RF4CE_USER_CONTROL_CODE_PLAY_FUNCTION
00092     = 0x60, // Play Mode - 1 byte
EMBER_AF_RF4CE_USER_CONTROL_CODE_PAUSE_PLAY_FUNCTION
00093     = 0x61,
EMBER_AF_RF4CE_USER_CONTROL_CODE_RECORD_FUNCTION
00094     = 0x62,
EMBER_AF_RF4CE_USER_CONTROL_CODE_PAUSE_RECORD_FUNCTION
00095     = 0x63,
EMBER_AF_RF4CE_USER_CONTROL_CODE_STOP_FUNCTION
00096     = 0x64,
EMBER_AF_RF4CE_USER_CONTROL_CODE_MUTE_FUNCTION
00097     = 0x65,
EMBER_AF_RF4CE_USER_CONTROL_CODE_RESTORE_VOLUME_FUNCTION
00098     = 0x66,
EMBER_AF_RF4CE_USER_CONTROL_CODE_TUNE_FUNCTION
00099     = 0x67, // Channel Identifier - 4 bytes
EMBER_AF_RF4CE_USER_CONTROL_CODE_SELECT_MEDIA_FUNCTION
00100     = 0x68, // UI Function Media - 1 byte
EMBER_AF_RF4CE_USER_CONTROL_CODE_SELECT_A_V_INPUT_FUNCTION
00101     = 0x69, // UI Function Select A/V Input - 1 byte
EMBER_AF_RF4CE_USER_CONTROL_CODE_SELECT_AUDIO_INPUT_FUNCTION
00102     = 0x6A, // UI Function Select Audio Input - 1 byte
EMBER_AF_RF4CE_USER_CONTROL_CODE_POWER_TOGGLE_FUNCTION
00103     = 0x6B,
EMBER_AF_RF4CE_USER_CONTROL_CODE_POWER_OFF_FUNCTION
00104     = 0x6C,
EMBER_AF_RF4CE_USER_CONTROL_CODE_POWER_ON_FUNCTION
00105     = 0x6D,
EMBER_AF_RF4CE_USER_CONTROL_CODE_F1_BLUE
00106     = 0x71,
EMBER_AF_RF4CE_USER_CONTROL_CODE_F2_RED
00107     = 0x72,
EMBER_AF_RF4CE_USER_CONTROL_CODE_F3_GREEN

```

```

00108     = 0x73,
EMBER_AF_RF4CE_USER_CONTROL_CODE_F4_YELLOW
00109     = 0x74,
EMBER_AF_RF4CE_USER_CONTROL_CODE_F5
00110     = 0x75,
EMBER_AF_RF4CE_USER_CONTROL_CODE_DATA
00111     = 0x76,
00111 };
00112
00116 typedef struct {
00117     uint8_t pairingIndex;
00118     EmberAfRf4ceZrcCommandCode commandCode;
00119     EmberAfRf4ceUserControlCode rcCommandCode
00120     ;
00120     const uint8_t *rcCommandPayload;
00121     uint8_t rcCommandPayloadLength;
00122     uint16_t timeMs;
00123 } EmberAfRf4ceZrcUserControlRecord;
00124
00128 #define EMBER_AF_RF4CE_ZRC_COMMANDS_SUPPORTED_SIZE 32
00129
00133 typedef struct {
00135     uint8_t contents[EMBER_AF_RF4CE_ZRC_COMMANDS_SUPPORTED_SIZE
00136 ];
00136 } EmberAfRf4ceZrcCommandsSupported;
00137
00138 #endif // __RF4CE_ZRC11_TYPES_H__

```

8.52 rf4ce-zrc11.h File Reference

```
#include "rf4ce-zrc11-types.h"
```

Functions

- EmberStatus **emberAfRf4ceZrc11Discovery** (EmberPanId panId, EmberNodeId nodeId, **EmberAfRf4ceDeviceType** searchDevType)
- EmberStatus **emberAfRf4ceZrc11EnableAutoDiscoveryResponse** (void)
- EmberStatus **emberAfRf4ceZrc11UserControlPress** (uint8_t pairingIndex, **EmberAfRf4ceUserControlCode** rcCommandCode, const uint8_t *rcCommandPayload, uint8_t rcCommandPayloadLength, bool atomic)
- EmberStatus **emberAfRf4ceZrc11UserControlRelease** (uint8_t pairingIndex, **EmberAfRf4ceUserControlCode** rcCommandCode)
- EmberStatus **emberAfRf4ceZrc11CommandDiscoveryRequest** (uint8_t pairingIndex)
- uint8_t * **emberAfRf4ceZrcCommandsSupportedContents** (**EmberAfRf4ceZrcCommandsSupported** *commandsSupported)

8.53 rf4ce-zrc11.h

```

00001 // Copyright 2014 Silicon Laboratories, Inc.
00002
00003 #ifndef __RF4CE_ZRC11_H__
00004 #define __RF4CE_ZRC11_H__
00005
00006 #include "rf4ce-zrc11-types.h"
00007
00009 #ifdef EMBER_AF_RF4CE_ZRC_IS_USER_CONTROL_ORIGINATOR
00100
00101     #define EMBER_AF_RF4CE_ZRC_IS_ORIGINATOR
00102 #endif
00103
00104 #ifdef EMBER_AF_RF4CE_ZRC_IS_USER_CONTROL_RECIPIENT
00105

```

```

00106 #define EMBER_AF_RF4CE_ZRC_IS_RECIPIENT
00107 #endif
00108
00127 EmberStatus emberAfRf4ceZrc11Discovery(EmberPanId
panId,
00128                                     EmberNodeId nodeId,
00129                                     EmberAfRf4ceDeviceType
searchDevType);
00130
00141 EmberStatus emberAfRf4ceZrc11EnableAutoDiscoveryResponse
(void);
00142
00165 EmberStatus emberAfRf4ceZrc11UserControlPress(
    uint8_t pairingIndex,
00166                                     EmberAfRf4ceUserControlCode
rcCommandCode,
00167                                     const uint8_t *rcCommandPayload,
00168                                     uint8_t rcCommandPayloadLength,
00169                                     bool atomic);
00170
00183 EmberStatus emberAfRf4ceZrc11UserControlRelease
(uint8_t pairingIndex,
00184                                     EmberAfRf4ceUserControlCode
rcCommandCode);
00185
00198 EmberStatus emberAfRf4ceZrc11CommandDiscoveryRequest
(uint8_t pairingIndex);
00199
00207 #ifdef DOXYGEN_SHOULD_SKIP_THIS
00208     uint8_t *emberAfRf4ceZrcCommandsSupportedContents
        (EmberAfRf4ceZrcCommandsSupported *
commandsSupported);
00209 #else
00210     #define emberAfRf4ceZrcCommandsSupportedContents(commandsSupported) \
00211         ((commandsSupported)->contents)
00212 #endif
00213
00214 #endif // __RF4CE_ZRC11_H__
00215
00216 // @} END addtogroup

```

8.54 rf4ce-zrc20-action-mapping-client.h File Reference

Functions

- void `emberAfRf4ceZrc20ActionMappingClientClearAllActionMappings` (void)
- EmberStatus `emberAfRf4ceZrc20ActionMappingClientClearActionMappingsPerPairing` (uint8_t pairingIndex)
- EmberStatus `emberAfRf4ceZrc20ActionMappingClientClearActionMapping` (uint8_t pairingIndex, EmberAfRf4ceDeviceType actionDeviceType, EmberAfRf4ceZrcActionBank actionBank, EmberAfRf4ceZrcActionCode actionCode)
- EmberStatus `emberAfRf4ceZrc20ActionMappingClientGetActionMapping` (uint8_t pairingIndex, EmberAfRf4ceDeviceType actionDeviceType, EmberAfRf4ceZrcActionBank actionBank, EmberAfRf4ceZrcActionCode actionCode, EmberAfRf4ceZrcActionMapping *actionMapping)
- EmberStatus `emberAfRf4ceZrc20ActionMappingClientLookUpActionMapping` (uint8_t pairingIndex, EmberAfRf4ceDeviceType actionDeviceType, EmberAfRf4ceZrcActionBank actionBank, EmberAfRf4ceZrcActionCode actionCode, EmberAfRf4ceZrcActionMapping *actionMapping)
- EmberStatus `emberAfRf4ceZrc20ActionMappingClientSetActionMapping` (uint8_t pairingIndex, EmberAfRf4ceDeviceType actionDeviceType, EmberAfRf4ceZrcActionBank actionBank, EmberAfRf4ceZrcActionCode actionCode, EmberAfRf4ceZrcActionMapping *actionMapping)

8.55 rf4ce-zrc20-action-mapping-client.h

```
00001 // Copyright 2014 Silicon Laboratories, Inc.
```

```

00002
00003 #ifndef __RF4CE_ZRC20_ACTION_MAPPING_CLIENT_H__
00004 #define __RF4CE_ZRC20_ACTION_MAPPING_CLIENT_H__
00005
00058 void emberAfRf4ceZrc20ActionMappingClientClearAllActionMappings
    (void);
00059
00068 EmberStatus emberAfRf4ceZrc20ActionMappingClientClearActionMappingsPerPairing
    (uint8_t pairingIndex);
00069
00084 EmberStatus emberAfRf4ceZrc20ActionMappingClientClearActionMapping
    (uint8_t pairingIndex,
00085     EmberAfRf4ceDeviceType actionDeviceType,
00086     EmberAfRf4ceZrcActionBank actionBank,
00087     EmberAfRf4ceZrcActionCode actionCode);
00088
00106 EmberStatus emberAfRf4ceZrc20ActionMappingClientGetActionMapping
    (uint8_t pairingIndex,
00107     EmberAfRf4ceDeviceType actionDeviceType,
00108     EmberAfRf4ceZrcActionBank actionBank,
00109     EmberAfRf4ceZrcActionCode actionCode,
00110     EmberAfRf4ceZrcActionMapping* actionMapping);
00111 #ifdef DOXYGEN_SHOULD_SKIP_THIS
00113
00134 EmberStatus emberAfRf4ceZrc20ActionMappingClientLookUpActionMapping
    (uint8_t pairingIndex,
00135     EmberAfRf4ceDeviceType actionDeviceType,
00136     EmberAfRf4ceZrcActionBank actionBank,
00137     EmberAfRf4ceZrcActionCode actionCode,
00138     EmberAfRf4ceZrcActionMapping* actionMapping);
00139 #else
00140     #define emberAfRf4ceZrc20ActionMappingClientLookUpActionMapping \
00141         emberAfRf4ceZrc20ActionMappingClientGetActionMapping
00142 #endif
00143
00161 EmberStatus emberAfRf4ceZrc20ActionMappingClientSetActionMapping
    (uint8_t pairingIndex,
00162     EmberAfRf4ceDeviceType actionDeviceType,
00163     EmberAfRf4ceZrcActionBank actionBank,
00164     EmberAfRf4ceZrcActionCode actionCode,
00165     EmberAfRf4ceZrcActionMapping* actionMapping);
00166
00167 #ifndef DOXYGEN_SHOULD_SKIP_THIS
00168
00169 /* @brief Set selected action mapping.
00170 *
00171 * @param pairingIndex The pairing index the set action mapping command
00172 * should be sent to.
00173 *
00174 * @param entryIndex The entry index of the action mapping to be set.
00175 *
00176 * @param actionMapping The action mapping structure describing the action
00177 * mapping.
00178 *
00179 * @return An ::EmberStatus value indicating whether the set attributes
00180 * command
00181 * was successfully sent out or the reason of failure.
00182 */
00182 EmberStatus emberAfRf4ceZrc20ActionMappingClientSetActionMapping(uint8_t
    pairingIndex,
00183                                         uint16_t
00184                                         entryIndex,
00184                                         EmberAfRf4ceZrcActionMapping* actionMapping);

```

```

00185
00186 /* @brief Clear selected action mapping on the client.
00187 *
00188 * @param pairingIndex The pairing index the clear action mapping command
00189 * should be sent to.
00190 *
00191 * @param entryIndex The entry index of the action mapping to be cleared.
00192 *
00193 * @return An ::EmberStatus value indicating whether the clear action mapping
00194 * command was successfully sent out or the reason of failure.
00195 */
00196 EmberStatus emAfRf4ceZrc20ActionMappingClientClearActionMapping(uint8_t
pairingIndex,
00197                                         uint16_t
entryIndex);

00198 /* @brief Get the number of mappable actions on the client.
00199 *
00200 * @return Mappable Action count
00201 */
00202 uint16_t emAfPluginRf4ceZrc20ActionMappingClientGetMappableActionCount(void);

00203 /* @brief Get mappable action at entryIndex.
00204 *
00205 * @param entryIndex The entry index of the mappable action to get.
00206 *
00207 * @param mappableAction The mappable action structure describing the
mappable
00208 * action.
00209 *
00210 * @return An ::EmberStatus value indicating whether the get mappable action
00211 * command was successfully sent out or the reason of failure.
00212 */
00213 EmberStatus emAfPluginRf4ceZrc20ActionMappingClientGetMappableAction(uint16_t
entryIndex,
00214
EmberAfRf4ceZrcMappableAction *mappableAction);
00215 #endif // DOXYGEN_SHOULD_SKIP_THIS
00216 #endif // __RF4CE_ZRC20_ACTION_MAPPING_CLIENT_H__
00217 // END addtogroup

```

8.56 rf4ce-zrc20-action-mapping-server.h File Reference

Functions

- EmberStatus [emberAfRf4ceZrc20ActionMappingServerRemapAction](#) ([EmberAfRf4ceZrcMappableAction](#) *mappableAction, [EmberAfRf4ceZrcActionMapping](#) *actionMapping)
- EmberStatus [emberAfRf4ceZrc20ActionMappingServerRestoreDefaultAction](#) ([EmberAfRf4ceZrcMappableAction](#) *mappableAction)
- void [emberAfRf4ceZrc20ActionMappingServerRestoreDefaultAllActions](#) (void)
- uint16_t [emberAfRf4ceZrc20ActionMappingServerGetMappableActionCount](#) (void)

8.57 rf4ce-zrc20-action-mapping-server.h

```

00001 // Copyright 2014 Silicon Laboratories, Inc.
00002
00003 #ifndef __RF4CE_ZRC20_ACTION_MAPPING_SERVER_H__
00004 #define __RF4CE_ZRC20_ACTION_MAPPING_SERVER_H__
00005
00058 EmberStatus emberAfRf4ceZrc20ActionMappingServerRemapAction
(EmberAfRf4ceZrcMappableAction* mappableAction,
00059
EmberAfRf4ceZrcActionMapping* actionMapping);
00060
00069 EmberStatus emberAfRf4ceZrc20ActionMappingServerRestoreDefaultAction

```

```

00070     (EmberAfRf4ceZrcMappableAction* mappableAction);
00074 void emberAfRf4ceZrc20ActionMappingServerRestoreDefaultAllActions
00075     (void);
00076
00080 uint16_t emberAfRf4ceZrc20ActionMappingServerGetMappableActionCount
00081     (void);
00082 #ifndef DOXYGEN_SHOULD_SKIP_THIS
00083
00084 /* @brief Clear entries in mapping table that point at serverEntryIndex.
00085 *
00086 * @param serverIndex The index of the mappable action on the server.
00087 *
00088 * @return An ::EmberStatus value indicating whether the clear mapping command
00089 * was successfully sent out or the reason of failure.
00090 */
00091 void emAfRf4ceZrc20ActionMappingServerClearMapping(uint16_t serverIndex);
00092
00093 /* @brief Clear all entries in mapping table.
00094 *
00095 */
00096 void emAfRf4ceZrc20ActionMappingServerClearAllMappings(void);
00097
00098 /* @brief Add entry in mapping table.
00099 *
00100 * @param pairingIndex The pairing index the add mapping command should be
00101 * sent to.
00102 *
00103 * @param entryIndex The entry index of the mappable action on the client.
00104 *
00105 * @param serverIndex The index of the mappable action on the server.
00106 *
00107 * @return An ::EmberStatus value indicating whether the add mapping command
00108 * was successfully sent out or the reason of failure.
00109 */
00110 EmberStatus emAfRf4ceZrc20ActionMappingServerUpdateOrAddMapping(uint8_t
00111     pairingIndex,
00112             uint16_t
00113     entryIndex,
00114             uint16_t
00115     serverIndex);
00116
00117 /* @brief Remove entry from mapping table.
00118 *
00119 * @param pairingIndex The pairing index the remove mapping command should be
00120 * sent to.
00121 *
00122 * @param entryIndex The entry index of the mappable action on the client.
00123 *
00124 * @return An ::EmberStatus value indicating whether the remove mapping
00125 * command
00126 * was successfully sent out or the reason of failure.
00127 */
00128 EmberStatus emAfRf4ceZrc20ActionMappingServerRemoveMapping(uint8_t pairingIndex
00129     ,
00130             uint16_t entryIndex)
00131 ;
00132
00133 /* @brief Get entry from mapping table.
00134 *
00135 * @param pairingIndex The pairing index the get mapping command should be
00136 * sent to.
00137 *
00138 * @param entryIndex The entry index of the mappable action on the client.
00139 *
00140 * @return An ::EmberStatus value indicating whether the get mapping command
00141 * was successfully sent out or the reason of failure.
00142 */
00143 EmberStatus emAfRf4ceZrc20ActionMappingServerGetMapping(uint8_t pairingIndex,
00144             uint16_t entryIndex,
00145             uint16_t *serverIndex);
00146
00147 /* @brief Clear mappable action at index of the table on the server.
00148 *
00149 * @param index The index the clear mappable action command should be sent
00150 * to.

```

```

00147 *
00148 * @return An ::EmberStatus value indicating whether the clear mappable action
00149 * command was successfully sent out or the reason of failure.
00150 */
00151 EmberStatus emAfRf4ceZrc20ActionMappingServerClearMappableAction(uint16_t index
00152 );
00152 /* @brief Clear all mappable actions on the server.
00153 *
00154 */
00156 void emAfRf4ceZrc20ActionMappingServerClearAllMappableActions(void);
00157
00158 /* @brief Set mappable action on the server.
00159 *
00160 * @param index The index of the mappable action to set.
00161 *
00162 * @param mappableAction The mappable action structure describing the
00163 * mappable
00164 * action to write to index location in the table.
00165 *
00166 * @return An ::EmberStatus value indicating whether the set mappable action
00167 * command was successfully sent out or the reason of failure.
00167 */
00168 EmberStatus emAfRf4ceZrc20ActionMappingServerSetMappableAction(uint16_t index,
00169
00170     EmberAfRf4ceZrcMappableAction *mappableAction);
00170
00171 /* @brief Get mappable action on the server.
00172 *
00173 * @param index The index of the mappable action to get.
00174 *
00175 * @param mappableAction The mappable action structure describing the
00176 * mappable
00177 * action to read from index location in the table.
00177 */
00178 *
00179 * @return An ::EmberStatus value indicating whether the get mappable action
00180 * command was successfully sent out or the reason of failure.
00180 */
00181 EmberStatus emAfRf4ceZrc20ActionMappingServerGetMappableAction(uint16_t index,
00182
00183     EmberAfRf4ceZrcMappableAction *mappableAction);
00183
00184 /* @brief Look up mappable action on the server.
00185 *
00186 * @param mappableAction Pointer to the mappable action structure describing
00187 * the mappable action to look up.
00188 *
00189 * @param index The index of the mappable action found.
00190 *
00191 * @return An ::EmberStatus value indicating whether the look up mappable
00192 * action command was successfully sent out or the reason of failure.
00193 */
00194 EmberStatus emAfRf4ceZrc20ActionMappingServerLookUpMappableAction(
00195     EmberAfRf4ceZrcMappableAction *mappableAction,
00195                                         uint16_t*
00196     index);
00196
00197 /* @brief Clear action mapping on the server.
00198 *
00199 * @param index The index of the action mapping to clear.
00200 *
00201 * @return An ::EmberStatus value indicating whether the clear action mapping
00202 * command was successfully sent out or the reason of failure.
00203 */
00204 EmberStatus emAfRf4ceZrc20ActionMappingServerClearActionMapping(uint16_t index)
00204 ;
00205
00206 /* @brief Clear all action mappings on the server.
00207 *
00208 */
00209 void emAfRf4ceZrc20ActionMappingServerClearAllActionMappings(void);
00210
00211 /* @brief Set action mapping on the server.
00212 *
00213 * @param index The index of the action mapping to set.
00214 *
00215 * @param actionMapping The action mapping structure describing the action
00216 * mapping to write to index location in the table.
00217 *
00218 * @return An ::EmberStatus value indicating whether the set action mapping

```

```

00219 * command was successfully sent out or the reason of failure.
00220 */
00221 EmberStatus emAfRf4ceZrc20ActionMappingServerSetActionMapping(uint16_t index,
00222     EmberAfRf4ceZrcActionMapping *actionMapping);
00223 /**
00224 * @brief Get action mapping on the server.
00225 *
00226 * @param index The index of the action mapping to get.
00227 *
00228 * @param actionMapping The action mapping structure describing the action
00229 * mapping to read from index location in the table.
00230 *
00231 * @return An ::EmberStatus value indicating whether the get action mapping
00232 * command was successfully sent out or the reason of failure.
00233 */
00234 EmberStatus emAfRf4ceZrc20ActionMappingServerGetActionMapping(uint16_t index,
00235     EmberAfRf4ceZrcActionMapping *actionMapping);
00236 #endif // DOXYGEN_SHOULD_SKIP_THIS
00238 #endif // __RF4CE_ZRC20_ACTION_MAPPING_SERVER_H__
00240 // END addtogroup

```

8.58 rf4ce-zrc20-action-mapping.h File Reference

Macros

- `#define ACTION_MAPPING_NEGOTIATION_PROCEDURE_DELAY_MSEC`
- `#define MAPPABLE_ACTION_RECORD_SIZE`

Functions

- `void emAfRf4ceZrcActionMappingBindingCompleteCallback (uint8_t pairingIndex)`
- `void emAfRf4ceZrcActionMappingStartNegotiationClient (uint8_t pairingIndex)`

8.58.1 Macro Definition Documentation

8.58.1.1 `#define ACTION_MAPPING_NEGOTIATION_PROCEDURE_DELAY_MSEC`

Definition at line 6 of file `rf4ce-zrc20-action-mapping.h`.

8.58.1.2 `#define MAPPABLE_ACTION_RECORD_SIZE`

Definition at line 11 of file `rf4ce-zrc20-action-mapping.h`.

8.58.2 Function Documentation

8.58.2.1 `void emAfRf4ceZrcActionMappingBindingCompleteCallback (uint8_t pairingIndex)`

8.58.2.2 `void emAfRf4ceZrcActionMappingStartNegotiationClient (uint8_t pairingIndex)`

8.59 rf4ce-zrc20-action-mapping.h

```

00001 // Copyright 2014 Silicon Laboratories, Inc.
00002
00003
00004 // We delay the negotiation procedure in case the GDP plugin is busy doing
00005 // something or the ZRC state machine is not in its initial state.
00006 #define ACTION_MAPPING_NEGOTIATION_PROCEDURE_DELAY_MSEC      500
00007
00008
00009 // Attribute ID (1 byte) + entry ID (2 bytes) + Attribute length (1 byte)
00010 // + Attribute value (3 bytes)
00011 #define MAPPABLE_ACTION_RECORD_SIZE                         7
00012
00013 void emAfRf4ceZrcActionMappingBindingCompleteCallback
    (uint8_t pairingIndex);
00014
00015 void emAfRf4ceZrcActionMappingStartNegotiationClient
    (uint8_t pairingIndex);

```

8.60 rf4ce-zrc20-attributes.h File Reference

Data Structures

- struct [EmAfRf4ceZrcAttributeDescriptor](#)
- struct [EmAfZrcBitmask](#)
- struct [EmAfZrcArrayedBitmask](#)
- struct [EmAfRf4ceZrcAttributes](#)

Macros

- #define [ZRC_BITMASK_SIZE](#)
- #define [APL_ZRC_PROFILE_VERSION_SIZE](#)
- #define [APL_ZRC_PROFILE_CAPABILITIES_SIZE](#)
- #define [APL_ACTION_REPEAT_TRIGGER_INTERVAL_SIZE](#)
- #define [APL_ACTION_REPEAT_WAIT_TIME_SIZE](#)
- #define [APL_ZRC_ACTION_BANKS_VERSION_SIZE](#)
- #define [APL_MAPPABLE_ACTIONS_SIZE](#)
- #define [APL_ZRC_PROFILE_VERSION_DEFAULT](#)
- #define [APL_ACTION_REPEAT_TRIGGER_INTERVAL_DEFAULT_MS](#)
- #define [APL_ACTION_REPEAT_WAIT_TIME_DEFAULT_MS](#)
- #define [APL_ZRC_ACTION_BANKS_VERSION_DEFAULT](#)
- #define [MAX_ZRC_ATTRIBUTE_SIZE](#)
- #define [IRBD_SUPPORT_ATTRIBUTES_COUNT](#)
- #define [ACTION_MAPPING_ATTRIBUTES_COUNT](#)
- #define [HA_ATTRIBUTES_COUNT](#)
- #define [ZRC_ATTRIBUTES_COUNT](#)
- #define [MAPPABLE_ACTION_ACTION_DEVICE_TYPE_OFFSET](#)
- #define [MAPPABLE_ACTION_ACTION_BANK_OFFSET](#)
- #define [MAPPABLE_ACTION_ACTION_CODE_OFFSET](#)
- #define [ACTION_MAPPING_FLAGS_OFFSET](#)
- #define [ACTION_MAPPING_IR_CONFIG_VENDOR_SPECIFIC_BIT](#)
- #define [HA_ATTRIBUTE_STATUS_OFFSET](#)
- #define [HA_ATTRIBUTE_STATUS_VALUE_AVAILABLE_FLAG](#)
- #define [HA_ATTRIBUTE_VALUE_OFFSET](#)

- #define ZRC_ATTRIBUTE_HAS_REMOTE_GET_ACCESS_BIT
- #define ZRC_ATTRIBUTE_HAS_REMOTE_SET_ACCESS_BIT
- #define ZRC_ATTRIBUTE_HAS_REMOTE_PUSH_ACCESS_BIT
- #define ZRC_ATTRIBUTE_HAS_REMOTE_PULL_ACCESS_BIT
- #define ZRC_ATTRIBUTE_IS_TWO_DIMENSIONAL_ARRAYED
- #define ZRC_ATTRIBUTE_LOCAL_NODE_SUPPORTED
- #define ZRC_ATTRIBUTE_REMOTE_NODE_SUPPORTED
- #define ZRC20_CAPABILITIES_NON_RESERVED_BITS_BITMASK
- #define LOCAL_NODE_SUPPORTS_ACTIONS_RECIPIENT_BIT
- #define LOCAL_NODE_SUPPORTS_ACTIONS_ORIGINATOR_BIT
- #define LOCAL_NODE_IS_AM_CLIENT_BIT
- #define LOCAL_NODE_SUPPORTS_VENDOR_SPECIFIC_IRDB_BIT
- #define LOCAL_NODE_IS_AM_SERVER_BIT
- #define LOCAL_NODE_INFORM_BIT
- #define LOCAL_NODE_IS_HA_ORIGINATOR_BIT
- #define LOCAL_NODE_IS_HA_RECIPIENT_BIT
- #define LOCAL_NODE_ZRC_CAPABILITIES
- #define emAfRf4ceZrcGetLocalNodeCapabilities()
- #define emAfRf4ceZrcGetRemoteNodeCapabilities(pairingIndex)
- #define emAfRf4ceZrcSetRemoteNodeCapabilities(pairingIndex, capabilities)
- #define emAfRf4ceZrcWriteLocalAttribute(attrId, entryIdOrValueLength, val)
- #define emAfRf4ceZrcReadLocalAttribute(attrId, entryId, val)
- #define emAfRf4ceZrcWriteRemoteAttribute(pairingIndex,attrId,entryIdOrValueLength,val)
- #define emAfRf4ceZrcReadRemoteAttribute(pairingIndex, attrId, entryId, val)

Functions

- uint8_t * emAfRf4ceZrcGetArrayedAttributePointer (uint8_t attrId, uint16_t entryId, uint8_t pairingIndex, uint8_t *index)
- void emAfRf4ceZrcReadOrWriteAttribute (uint8_t pairingIndex, uint8_t attrId, uint16_t entryIdOrValueLength, bool isRead, uint8_t *val)

Variables

- EmAfRf4ceZrcAttributes emAfRf4ceZrcLocalNodeAttributes
- EmAfRf4ceZrcAttributes emAfRf4ceZrcRemoteNodeAttributes

8.60.1 Macro Definition Documentation

8.60.1.1 #define ZRC_BITMASK_SIZE

Definition at line 3 of file rf4ce-zrc20-attributes.h.

8.60.1.2 #define APL_ZRC_PROFILE_VERSION_SIZE

Definition at line 6 of file rf4ce-zrc20-attributes.h.

8.60.1.3 #define APL_ZRC_PROFILE_CAPABILITIES_SIZE

Definition at line 7 of file [rf4ce-zrc20-attributes.h](#).

8.60.1.4 #define APL_ACTION_REPEAT_TRIGGER_INTERVAL_SIZE

Definition at line 8 of file [rf4ce-zrc20-attributes.h](#).

8.60.1.5 #define APL_ACTION_REPEAT_WAIT_TIME_SIZE

Definition at line 9 of file [rf4ce-zrc20-attributes.h](#).

8.60.1.6 #define APL_ZRC_ACTION_BANKS_VERSION_SIZE

Definition at line 10 of file [rf4ce-zrc20-attributes.h](#).

8.60.1.7 #define APL_MAPPABLE_ACTIONS_SIZE

Definition at line 13 of file [rf4ce-zrc20-attributes.h](#).

8.60.1.8 #define APL_ZRC_PROFILE_VERSION_DEFAULT

Definition at line 16 of file [rf4ce-zrc20-attributes.h](#).

8.60.1.9 #define APL_ACTION_REPEAT_TRIGGER_INTERVAL_DEFAULT_MS

Definition at line 17 of file [rf4ce-zrc20-attributes.h](#).

8.60.1.10 #define APL_ACTION_REPEAT_WAIT_TIME_DEFAULT_MS

Definition at line 18 of file [rf4ce-zrc20-attributes.h](#).

8.60.1.11 #define APL_ZRC_ACTION_BANKS_VERSION_DEFAULT

Definition at line 19 of file [rf4ce-zrc20-attributes.h](#).

8.60.1.12 #define MAX_ZRC_ATTRIBUTE_SIZE

Definition at line 21 of file [rf4ce-zrc20-attributes.h](#).

8.60.1.13 #define IRBD_SUPPORT_ATTRIBUTES_COUNT

Definition at line 28 of file [rf4ce-zrc20-attributes.h](#).

8.60.1.14 #define ACTION_MAPPING_ATTRIBUTES_COUNT

Definition at line 34 of file [rf4ce-zrc20-attributes.h](#).

8.60.1.15 #define HA_ATTRIBUTES_COUNT

Definition at line 40 of file [rf4ce-zrc20-attributes.h](#).

8.60.1.16 #define ZRC_ATTRIBUTES_COUNT

Definition at line 43 of file [rf4ce-zrc20-attributes.h](#).

8.60.1.17 #define MAPPABLE_ACTION_ACTION_DEVICE_TYPE_OFFSET

Definition at line 48 of file [rf4ce-zrc20-attributes.h](#).

8.60.1.18 #define MAPPABLE_ACTION_ACTION_BANK_OFFSET

Definition at line 49 of file [rf4ce-zrc20-attributes.h](#).

8.60.1.19 #define MAPPABLE_ACTION_ACTION_CODE_OFFSET

Definition at line 50 of file [rf4ce-zrc20-attributes.h](#).

8.60.1.20 #define ACTION_MAPPING_FLAGS_OFFSET

Definition at line 52 of file [rf4ce-zrc20-attributes.h](#).

8.60.1.21 #define ACTION_MAPPING_IR_CONFIG_VENDOR_SPECIFIC_BIT

Definition at line 53 of file [rf4ce-zrc20-attributes.h](#).

8.60.1.22 #define HA_ATTRIBUTE_STATUS_OFFSET

Definition at line 55 of file [rf4ce-zrc20-attributes.h](#).

8.60.1.23 #define HA_ATTRIBUTE_STATUS_VALUE_AVAILABLE_FLAG

Definition at line 56 of file [rf4ce-zrc20-attributes.h](#).

8.60.1.24 #define HA_ATTRIBUTE_VALUE_OFFSET

Definition at line 57 of file [rf4ce-zrc20-attributes.h](#).

8.60.1.25 `#define ZRC_ATTRIBUTE_HAS_REMOTE_GET_ACCESS_BIT`

Definition at line 60 of file [rf4ce-zrc20-attributes.h](#).

8.60.1.26 `#define ZRC_ATTRIBUTE_HAS_REMOTE_SET_ACCESS_BIT`

Definition at line 61 of file [rf4ce-zrc20-attributes.h](#).

8.60.1.27 `#define ZRC_ATTRIBUTE_HAS_REMOTE_PUSH_ACCESS_BIT`

Definition at line 62 of file [rf4ce-zrc20-attributes.h](#).

8.60.1.28 `#define ZRC_ATTRIBUTE_HAS_REMOTE_PULL_ACCESS_BIT`

Definition at line 63 of file [rf4ce-zrc20-attributes.h](#).

8.60.1.29 `#define ZRC_ATTRIBUTE_IS_TWO_DIMENSIONAL_ARRAYED`

Definition at line 64 of file [rf4ce-zrc20-attributes.h](#).

8.60.1.30 `#define ZRC_ATTRIBUTE_LOCAL_NODE_SUPPORTED`

Definition at line 65 of file [rf4ce-zrc20-attributes.h](#).

8.60.1.31 `#define ZRC_ATTRIBUTE_REMOTE_NODE_SUPPORTED`

Definition at line 66 of file [rf4ce-zrc20-attributes.h](#).

8.60.1.32 `#define ZRC20_CAPABILITIES_NON_RESERVED_BITS_BITMASK`

Definition at line 103 of file [rf4ce-zrc20-attributes.h](#).

8.60.1.33 `#define LOCAL_NODE_SUPPORTS_ACTIONS_RECIPIENT_BIT`

Definition at line 109 of file [rf4ce-zrc20-attributes.h](#).

8.60.1.34 `#define LOCAL_NODE_SUPPORTS_ACTIONS_ORIGINATOR_BIT`

Definition at line 115 of file [rf4ce-zrc20-attributes.h](#).

8.60.1.35 `#define LOCAL_NODE_IS_AM_CLIENT_BIT`

Definition at line 126 of file [rf4ce-zrc20-attributes.h](#).

8.60.1.36 #define LOCAL_NODE_SUPPORTS_VENDOR_SPECIFIC_IRDB_BIT

Definition at line 127 of file [rf4ce-zrc20-attributes.h](#).

8.60.1.37 #define LOCAL_NODE_IS_AM_SERVER_BIT

Definition at line 133 of file [rf4ce-zrc20-attributes.h](#).

8.60.1.38 #define LOCAL_NODE_INFORM_BIT

Definition at line 139 of file [rf4ce-zrc20-attributes.h](#).

8.60.1.39 #define LOCAL_NODE_IS_HA_ORIGINATOR_BIT

Definition at line 145 of file [rf4ce-zrc20-attributes.h](#).

8.60.1.40 #define LOCAL_NODE_IS_HA_RECIPIENT_BIT

Definition at line 151 of file [rf4ce-zrc20-attributes.h](#).

8.60.1.41 #define LOCAL_NODE_ZRC_CAPABILITIES

Definition at line 154 of file [rf4ce-zrc20-attributes.h](#).

8.60.1.42 #define emAfRf4ceZrcGetLocalNodeCapabilities()

Definition at line 169 of file [rf4ce-zrc20-attributes.h](#).

8.60.1.43 #define emAfRf4ceZrcGetRemoteNodeCapabilities(pairingIndex)

Definition at line 172 of file [rf4ce-zrc20-attributes.h](#).

8.60.1.44 #define emAfRf4ceZrcSetRemoteNodeCapabilities(pairingIndex, capabilities)

Definition at line 177 of file [rf4ce-zrc20-attributes.h](#).

8.60.1.45 #define emAfRf4ceZrcWriteLocalAttribute(attrId, entryIdOrValueLength, val)

Definition at line 203 of file [rf4ce-zrc20-attributes.h](#).

8.60.1.46 #define emAfRf4ceZrcReadLocalAttribute(attrId, entryId, val)

Definition at line 210 of file [rf4ce-zrc20-attributes.h](#).

8.60.1.47 #define emAfRf4ceZrcWriteRemoteAttribute(pairingIndex, attrId, entryIdOrValueLength, val)

Definition at line 217 of file [rf4ce-zrc20-attributes.h](#).

8.60.1.48 #define emAfRf4ceZrcReadRemoteAttribute(pairingIndex, attrId, entryId, val)

Definition at line 227 of file [rf4ce-zrc20-attributes.h](#).

8.60.2 Function Documentation

8.60.2.1 uint8_t* emAfRf4ceZrcGetArrayedAttributePointer (uint8_t attrId, uint16_t entryId, uint8_t pairingIndex, uint8_t * index)

8.60.2.2 void emAfRf4ceZrcReadOrWriteAttribute (uint8_t pairingIndex, uint8_t attrId, uint16_t entryIdOrValueLength, bool isRead, uint8_t * val)

8.60.3 Variable Documentation

8.60.3.1 EmAfRf4ceZrcAttributes emAfRf4ceZrcLocalNodeAttributes

8.60.3.2 EmAfRf4ceZrcAttributes emAfRf4ceZrcRemoteNodeAttributes

8.61 rf4ce-zrc20-attributes.h

```

00001 // Copyright 2014 Silicon Laboratories, Inc.
00002
00003 #define ZRC_BITMASK_SIZE 32
00004
00005 // Fixed size ZRC attributes size (scalar)
00006 #define APL_ZRC_PROFILE_VERSION_SIZE 2
00007 #define APL_ZRC_PROFILE_CAPABILITIES_SIZE 4
00008 #define APL_ACTION_REPEAT_TRIGGER_INTERVAL_SIZE 1
00009 #define APL_ACTION_REPEAT_WAIT_TIME_SIZE 2
00010 #define APL_ZRC_ACTION_BANKS_VERSION_SIZE 2
00011
00012 // Fixed size ZRC attributes size (arrayed)
00013 #define APL_MAPPABLE_ACTIONS_SIZE 3
00014
00015 // Default values for those that are defined.
00016 #define APL_ZRC_PROFILE_VERSION_DEFAULT 0x0200
00017 #define APL_ACTION_REPEAT_TRIGGER_INTERVAL_DEFAULT_MS \
    (APLC_MAX_ACTION_REPEAT_TRIGGER_INTERVAL_MS / 2)
00018 #define APL_ACTION_REPEAT_WAIT_TIME_DEFAULT_MS \
    (APLC_MAX_ACTION_REPEAT_TRIGGER_INTERVAL_MS * 2)
00019 #define APL_ZRC_ACTION_BANKS_VERSION_DEFAULT 0x0100
00020
00021 #define MAX_ZRC_ATTRIBUTE_SIZE \
    EMBER_AF_RF4CE_MAXIMUM_RF4CE_PAYLOAD_LENGTH
00022
00023 #if (defined(EMBER_AF_PLUGIN_RF4CE_ZRC20_LOCAL_IRDB_VENDOR_ATTRIBUTE_SUPPORT) \
    || \
    defined(EMBER_AF_PLUGIN_RF4CE_ZRC20_REMOTE_IRDB_VENDOR_ATTRIBUTES_SUPPORT) \
    || \
    defined(EMBER_SCRIPTED_TEST))
00024 #define IRBD_SUPPORT_ATTRIBUTES_COUNT 1
00025 #else
00026 #define IRBD_SUPPORT_ATTRIBUTES_COUNT 0
00027 #endif // EMBER_AF_PLUGIN_RF4CE_ZRC20_LOCAL_IRDB_VENDOR_ATTRIBUTE_SUPPORT || \
    EMBER_AF_PLUGIN_RF4CE_ZRC20_REMOTE_IRDB_VENDOR_ATTRIBUTES_SUPPORT
00030
00031 #if defined(EMBER_AF_PLUGIN_RF4CE_ZRC20_IS_ACTION_MAPPING_CLIENT) || \
    defined(EMBER_AF_PLUGIN_RF4CE_ZRC20_IS_ACTION_MAPPING_SERVER) || \
    defined(EMBER_SCRIPTED_TEST)
00032 #define ACTION_MAPPING_ATTRIBUTES_COUNT 2

```

```

00033 #else
00034 #define ACTION_MAPPING_ATTRIBUTES_COUNT 0
00035 #endif // EMBER_AF_RF4CE_ZRC_IS_HA_ACTIONS_ORIGINATOR || 
    EMBER_AF_PLUGIN_RF4CE_ZRC20_IS_ACTION_MAPPING_SERVER
00036
00037 #if defined(EMBER_AF_RF4CE_ZRC_IS_HA_ACTIONS_ORIGINATOR) || defined
    (EMBER_AF_RF4CE_ZRC_IS_HA_ACTIONS_RECIPIENT) || defined(EMBER_SCRIPTED_TEST)
00038 #define HA_ATTRIBUTES_COUNT 2
00039 #else
00040 #define HA_ATTRIBUTES_COUNT 0
00041 #endif // EMBER_AF_RF4CE_ZRC_IS_HA_ACTIONS_ORIGINATOR || 
    EMBER_AF_RF4CE_ZRC_IS_HA_ACTIONS_RECIPIENT
00042
00043 #define ZRC_ATTRIBUTES_COUNT (9 + IRBD_SUPPORT_ATTRIBUTES_COUNT
\ 
00044 \
+ ACTION_MAPPING_ATTRIBUTES_COUNT
\
00045 \
+ HA_ATTRIBUTES_COUNT)
00046
00047 // Add here all the other mappable action and action mapping defines as needed.
00048 #define MAPPABLE_ACTION_DEVICE_TYPE_OFFSET 0
00049 #define MAPPABLE_ACTION_ACTION_BANK_OFFSET 1
00050 #define MAPPABLE_ACTION_ACTION_CODE_OFFSET 2
00051
00052 #define ACTION_MAPPING_FLAGS_OFFSET 0
00053 #define ACTION_MAPPING_IR_CONFIG_VENDOR_SPECIFIC_BIT 0x01
00054
00055 #define HA_ATTRIBUTE_STATUS_OFFSET 0
00056 #define HA_ATTRIBUTE_STATUS_VALUE_AVAILABLE_FLAG 0x01
00057 #define HA_ATTRIBUTE_VALUE_OFFSET 1
00058
00059 // Attribute descriptor bitmask field definitions.
00060 #define ZRC_ATTRIBUTE_HAS_REMOTE_GET_ACCESS_BIT 0x01
00061 #define ZRC_ATTRIBUTE_HAS_REMOTE_SET_ACCESS_BIT 0x02
00062 #define ZRC_ATTRIBUTE_HAS_REMOTE_PUSH_ACCESS_BIT 0x04
00063 #define ZRC_ATTRIBUTE_HAS_REMOTE_PULL_ACCESS_BIT 0x08
00064 #define ZRC_ATTRIBUTE_IS_TWO_DIMENSIONAL_ARRAYED 0x10
00065 #define ZRC_ATTRIBUTE_LOCAL_NODE_SUPPORTED 0x20
00066 #define ZRC_ATTRIBUTE_REMOTE_NODE_SUPPORTED 0x40
00067
00068 typedef struct {
00069     uint8_t id;
00070     uint8_t size;
00071     uint8_t bitmask;
00072 } EmAfRf4ceZrcAttributeDescriptor;
00073
00074 typedef struct {
00075     uint8_t contents[ZRC_BITMASK_SIZE];
00076 } EmAfZrcBitmask;
00077
00078 typedef struct {
00079     bool inUse;
00080     uint8_t entryId;
00081     uint8_t contents[ZRC_BITMASK_SIZE];
00082 } EmAfZrcArrayedBitmask;
00083
00084 typedef struct {
00085     // Scalar
00086     uint16_t zrcProfileVersion;
00087     uint16_t zrcActionBanksVersion;
00088     EmAfZrcBitmask *actionBanksSupportedRx;
00089     EmAfZrcBitmask *actionBanksSupportedTx;
00090 #if (defined(EMBER_AF_PLUGIN_RF4CE_ZRC20_LOCAL_IRDB_VENDOR_ATTRIBUTE_SUPPORT) \
\ 
00091     || 
        defined(EMBER_AF_PLUGIN_RF4CE_ZRC20_REMOTE_IRDB_VENDOR_ATTRIBUTES_SUPPORT) \
00092     || defined(EMBER_SCRIPTED_TEST))
00093     uint16_t *IRDBVendorSupport;
00094 #endif
00095
00096     // Arrayed
00097     EmAfZrcArrayedBitmask *actionCodesSupportedRx
00098 ;
00099     EmAfZrcArrayedBitmask *actionCodesSupportedTx
00100 ;
00101 // Mappable actions, action mappings and HA attributes are stored separately.
00101 } EmAfRf4ceZrcAttributes;
00102
00103 #define ZRC20_CAPABILITIES_NON_RESERVED_BITS_BITMASK 0x000000FF

```

```

00104
00105 // Pre-processing build up of the local node ZRC capabilities.
00106 #if defined(EMBER_AF_RF4CE_ZRC_ACTION_BANKS_RX)
00107 #define LOCAL_NODE_SUPPORTS_ACTIONS_RECIPIENT_BIT
    EMBER_AF_RF4CE_ZRC_CAPABILITY_SUPPORT_ACTIONS_RECIPIENT
00108 #else
00109 #define LOCAL_NODE_SUPPORTS_ACTIONS_RECIPIENT_BIT 0
00110 #endif // EMBER_AF_RF4CE_ZRC_ACTION_BANKS_RX
00111
00112 #if defined(EMBER_AF_RF4CE_ZRC_ACTION_BANKS_TX)
00113 #define LOCAL_NODE_SUPPORTS_ACTIONS_ORIGINATOR_BIT
    EMBER_AF_RF4CE_ZRC_CAPABILITY_SUPPORT_ACTIONS_ORIGINATOR
00114 #else
00115 #define LOCAL_NODE_SUPPORTS_ACTIONS_ORIGINATOR_BIT 0
00116 #endif // EMBER_AF_RF4CE_ZRC_ACTION_BANKS_RX
00117
00118 #if defined(EMBER_AF_PLUGIN_RF4CE_ZRC20_IS_ACTION_MAPPING_CLIENT)
00119 #define LOCAL_NODE_IS_AM_CLIENT_BIT
    EMBER_AF_RF4CE_ZRC_CAPABILITY_SUPPORT_ACTION_MAPPING_CLIENT
00120 #if defined(EMBER_AF_PLUGIN_RF4CE_ZRC20_LOCAL_IRDB_VENDOR_ATTRIBUTE_SUPPORT)
00121 #define LOCAL_NODE_SUPPORTS_VENDOR_SPECIFIC_IRDB_BIT
    EMBER_AF_RF4CE_ZRC_CAPABILITY_SUPPORT_VENDOR_SPECIFIC_IRDB_FORMATS
00122 #else
00123 #define LOCAL_NODE_SUPPORTS_VENDOR_SPECIFIC_IRDB_BIT 0
00124 #endif // EMBER_AF_PLUGIN_RF4CE_ZRC20_LOCAL_IRDB_VENDOR_ATTRIBUTE_SUPPORT
00125 #else
00126 #define LOCAL_NODE_IS_AM_CLIENT_BIT 0
00127 #define LOCAL_NODE_SUPPORTS_VENDOR_SPECIFIC_IRDB_BIT 0
00128 #endif // EMBER_AF_PLUGIN_RF4CE_ZRC20_IS_ACTION_MAPPING_CLIENT
00129
00130 #if defined(EMBER_AF_PLUGIN_RF4CE_ZRC20_IS_ACTION_MAPPING_SERVER)
00131 #define LOCAL_NODE_IS_AM_SERVER_BIT
    EMBER_AF_RF4CE_ZRC_CAPABILITY_SUPPORT_ACTION_MAPPING_SERVER
00132 #else
00133 #define LOCAL_NODE_IS_AM_SERVER_BIT 0
00134 #endif // EMBER_AF_PLUGIN_RF4CE_ZRC20_IS_ACTION_MAPPING_SERVER
00135
00136 #if defined(EMBER_AF_PLUGIN_RF4CE_ZRC20_INFORM_ABOUT_SUPPORTED_ACTIONS)
00137 #define LOCAL_NODE_INFORM_BIT
    EMBER_AF_RF4CE_ZRC_CAPABILITY_INFORM_ABOUT_SUPPORTED_ACTIONS
00138 #else
00139 #define LOCAL_NODE_INFORM_BIT 0
00140 #endif // EMBER_AF_PLUGIN_RF4CE_ZRC20_INFORM_ABOUT_SUPPORTED_ACTIONS
00141
00142 #if defined(EMBER_AF_RF4CE_ZRC_IS_HA_ACTIONS_ORIGINATOR)
00143 #define LOCAL_NODE_IS_HA_ORIGINATOR_BIT
    EMBER_AF_RF4CE_ZRC_CAPABILITY_SUPPORT_HA_ACTIONS_ORIGINATOR
00144 #else
00145 #define LOCAL_NODE_IS_HA_ORIGINATOR_BIT 0
00146 #endif // EMBER_AF_RF4CE_ZRC_IS_HA_ACTIONS_ORIGINATOR
00147
00148 #if defined(EMBER_AF_RF4CE_ZRC_IS_HA_ACTIONS_RECIPIENT)
00149 #define LOCAL_NODE_IS_HA_RECIPIENT_BIT
    EMBER_AF_RF4CE_ZRC_CAPABILITY_SUPPORT_HA_ACTIONS_RECIPIENT
00150 #else
00151 #define LOCAL_NODE_IS_HA_RECIPIENT_BIT 0
00152 #endif // EMBER_AF_RF4CE_ZRC_IS_HA_ACTIONS_RECIPIENT
00153
00154 #define LOCAL_NODE_ZRC_CAPABILITIES
    \
00155     (0x00000000
00156     \
        | LOCAL_NODE_SUPPORTS_ACTIONS_RECIPIENT_BIT
00157     \
        | LOCAL_NODE_SUPPORTS_ACTIONS_ORIGINATOR_BIT
00158     \
        | LOCAL_NODE_IS_AM_CLIENT_BIT
00159     \
        | LOCAL_NODE_IS_AM_SERVER_BIT
00160     \
        | LOCAL_NODE_SUPPORTS_VENDOR_SPECIFIC_IRDB_BIT
00161     \
        | LOCAL_NODE_INFORM_BIT
00162     \
        | LOCAL_NODE_IS_HA_ORIGINATOR_BIT
00163     \
        | LOCAL_NODE_IS_HA_RECIPIENT_BIT)
00164
00165 #if defined(EMBER_SCRIPTED_TEST)
00166 extern uint32_t localNodeZrcCapabilities;

```

```

00167 #define emAfrf4ceZrcGetLocalNodeCapabilities() (localNodeZrcCapabilities)
00168 #else
00169 #define emAfrf4ceZrcGetLocalNodeCapabilities() (LOCAL_NODE_ZRC_CAPABILITIES)
00170 #endif
00171
00172 #define emAfrf4ceZrcGetRemoteNodeCapabilities(pairingIndex)
00173   \
00174   ((uint32_t)((emAfrf4ceZrcGetRemoteNodeFlags(pairingIndex)
00175   \
00176   & ZRC_INTERNAL_FLAGS_CAPABILITIES_MASK)
00177   \
00178   >> ZRC_INTERNAL_FLAGS_CAPABILITIES_OFFSET))
00179
00180 #define emAfrf4ceZrcSetRemoteNodeCapabilities(pairingIndex, capabilities)
00181   \
00182   emAfrf4ceZrcSetRemoteNodeFlags((pairingIndex),
00183   \
00184   ((emAfrf4ceZrcGetRemoteNodeFlags(pairingIndex)
00185   \
00186   & ~ZRC_INTERNAL_FLAGS_CAPABILITIES_MASK)
00187   \
00188   | ((capabilities & ZRC20_CAPABILITIES_NON_RESERVED_BITS_BITMASK)
00189   \
00190   << ZRC_INTERNAL_FLAGS_CAPABILITIES_OFFSET)))
00191
00192 extern EmAfrf4ceZrcAttributes
00193 emAfrf4ceZrcLocalNodeAttributes;
00194 extern EmAfrf4ceZrcAttributes
00195 emAfrf4ceZrcRemoteNodeAttributes;
00196
00197 // This API returns a pointer to the requested (attribute ID, entry ID) pair
00198 // if it exists. Otherwise it returns NULL. The pairing index is used to
00199 // distinguish between local and remote attributes. If the pairing index is
00200 // 0xFF, then the API will look at the local attributes, otherwise it will look
00201 // at the remote attributes corresponding to the passed pairing index.
00202 uint8_t *emAfrf4ceZrcGetArrayedAttributePointer
00203 (uint8_t attrId,
00204          uint16_t entryId,
00205          uint8_t pairingIndex,
00206          uint8_t *index);
00207
00208 void emAfrf4ceZrcReadOrWriteAttribute(uint8_t
00209 pairingIndex,
00210          uint8_t attrId,
00211          uint16_t entryIdOrValueLength,
00212          bool isRead,
00213          uint8_t *val);
00214
00215 #define emAfrf4ceZrcWriteLocalAttribute(attrId, entryIdOrValueLength, val)
00216   \
00217   emAfrf4ceZrcReadOrWriteAttribute(0xFF,
00218   \
00219   (attrId),
00220   \
00221   (entryIdOrValueLength),
00222   \
00223   false,
00224   \
00225   (uint8_t*) (val))
00226
00227 #define emAfrf4ceZrcReadLocalAttribute(attrId, entryId, val)
00228   \
00229   emAfrf4ceZrcReadOrWriteAttribute(0xFF,
00230   \
00231   (attrId),
00232   \
00233   (entryId),
00234   \
00235   true,
00236   \
00237   (uint8_t*) (val))
00238
00239 #define emAfrf4ceZrcWriteRemoteAttribute(pairingIndex,
00240   \
00241   attrId,
00242   \
00243   entryIdOrValueLength,
00244   \
00245   val)
00246

```

```

00221     emAfRf4ceZrcReadOrWriteAttribute((pairingIndex),
00222             (attrId),
00223             (entryIdOrValueLength),
00224             false,
00225             (uint8_t*) (val))
00226
00227 #define emAfRf4ceZrcReadRemoteAttribute(pairingIndex, attrId, entryId, val)
00228     emAfRf4ceZrcReadOrWriteAttribute((pairingIndex),
00229             (attrId),
00230             (entryId),
00231             true,
00232             (uint8_t*) (val))

```

8.62 rf4ce-zrc20-ha-actions.h File Reference

Macros

- `#define HA_ACTIONS_ANNOUNCEMENT_PROCEDURE_DELAY_MSEC`
- `#define HA_SUPPORTED_MAX_INSTANCE_ID_INDEX`

Functions

- `void emAfRf4ceZrcHAActionsBindingCompleteCallback (uint8_t pairingIndex)`
- `void emAfRf4ceZrcStartHomeAutomationAnnouncementClient (uint8_t pairingIndex)`

8.62.1 Macro Definition Documentation

8.62.1.1 `#define HA_ACTIONS_ANNOUNCEMENT_PROCEDURE_DELAY_MSEC`

Definition at line 6 of file [rf4ce-zrc20-ha-actions.h](#).

8.62.1.2 `#define HA_SUPPORTED_MAX_INSTANCE_ID_INDEX`

Definition at line 9 of file [rf4ce-zrc20-ha-actions.h](#).

8.62.2 Function Documentation

8.62.2.1 `void emAfRf4ceZrcHAActionsBindingCompleteCallback (uint8_t pairingIndex)`

8.62.2.2 `void emAfRf4ceZrcStartHomeAutomationAnnouncementClient (uint8_t pairingIndex)`

8.63 rf4ce-zrc20-ha-actions.h

```

00001 // Copyright 2014 Silicon Laboratories, Inc.
00002
00003 // We delay the announcement procedure in case the GDP plugin is busy doing
00004

```

```

00005 // something or the ZRC state machine is not in its initial state.
00006 #define HA_ACTIONS_ANNOUNCEMENT_PROCEDURE_DELAY_MSEC      500
00007
00008 // Action bank 0x80-0x9F
00009 #define HA_SUPPORTED_MAX_INSTANCE_ID_INDEX   31
00010
00011 void emAfRf4ceZrcHAActionsBindingCompleteCallback
    (uint8_t pairingIndex);
00012
00013 void emAfRf4ceZrcStartHomeAutomationAnnouncementClient
    (uint8_t pairingIndex);

```

8.64 rf4ce-zrc20-ha-client.h File Reference

Macros

- `#define emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene(scene)`
- `#define emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene(scene)`
- `#define emberAfRf4ceZrcHaFillCommandPreviousDestinationGroup()`
- `#define emberAfRf4ceZrcHaFillCommandNextDestinationGroup()`

Enumerations

- `enum EmberAfThermostatSetpointRaiseLowerMode { EMBER_ZCL_THERMOSTAT_SETPOINT_RAISE_LOWER_MODE_HEAT, EMBER_ZCL_THERMOSTAT_SETPOINT_RAISE_LOWER_MODE_COOL, EMBER_ZCL_THERMOSTAT_SETPOINT_RAISE_LOWER_MODE_BOOTH }`

Functions

- `uint16_t emAfPluginRf4ceZrc20HaFillExternalBuffer (PGM_P format,...)`
- `EmberStatus emberAfRf4ceZrcHaSend (uint8_t pairingIndex, uint8_t haInstanceId)`

Scenes Commands

- `#define emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene0()`
- `#define emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene1()`
- `#define emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene2()`
- `#define emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene3()`
- `#define emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene4()`
- `#define emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene5()`
- `#define emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene6()`
- `#define emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene7()`
- `#define emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene8()`
- `#define emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene9()`
- `#define emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene10()`
- `#define emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene11()`
- `#define emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene12()`
- `#define emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene13()`
- `#define emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene14()`
- `#define emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene15()`
- `#define emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene0()`

- #define `emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene1()`
- #define `emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene2()`
- #define `emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene3()`
- #define `emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene4()`
- #define `emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene5()`
- #define `emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene6()`
- #define `emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene7()`
- #define `emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene8()`
- #define `emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene9()`
- #define `emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene10()`
- #define `emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene11()`
- #define `emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene12()`
- #define `emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene13()`
- #define `emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene14()`
- #define `emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene15()`

On/off Commands

- #define `emberAfRf4ceZrcHaFillCommandOnOffClusterOff()`
- #define `emberAfRf4ceZrcHaFillCommandOnOffClusterOn()`
- #define `emberAfRf4ceZrcHaFillCommandOnOffClusterToggle()`

Level Control Commands

- #define `emberAfRf4ceZrcHaFillCommandLevelControlClusterMoveToLevel(level,transitionTime)`
- #define `emberAfRf4ceZrcHaFillCommandLevelControlClusterMove(moveMode,rate)`
- #define `emberAfRf4ceZrcHaFillCommandLevelControlClusterStep(stepMode,stepSize,transitionTime)`
- #define `emberAfRf4ceZrcHaFillCommandLevelControlClusterStop()`
- #define `emberAfRf4ceZrcHaFillCommandLevelControlClusterMoveToLevelWithOnOff(level,transitionTime)`
- #define `emberAfRf4ceZrcHaFillCommandLevelControlClusterMoveWithOnOff(moveMode, rate)`
- #define `emberAfRf4ceZrcHaFillCommandLevelControlClusterStepWithOnOff(stepMode,stepSize,transitionTime)`
- #define `emberAfRf4ceZrcHaFillCommandLevelControlClusterStopWithOnOff()`

Door Lock Commands

- #define `emberAfRf4ceZrcHaFillCommandDoorLockClusterLockDoor(pinRfidCode)`
- #define `emberAfRf4ceZrcHaFillCommandDoorLockClusterUnlockDoor(pinRfidCode)`
- #define `emberAfRf4ceZrcHaFillCommandDoorLockClusterToggle(pinRfidCode)`
- #define `emberAfRf4ceZrcHaFillCommandDoorLockClusterUnlockWithTimeout(timeoutInSeconds,pinRfidCode)`

Window Covering Commands

- #define `emberAfRf4ceZrcHaFillCommandWindowCoveringClusterUpOpen()`
- #define `emberAfRf4ceZrcHaFillCommandWindowCoveringClusterDownClose()`
- #define `emberAfRf4ceZrcHaFillCommandWindowCoveringClusterStop()`
- #define `emberAfRf4ceZrcHaFillCommandWindowCoveringClusterGoToLiftValue(liftValue)`
- #define `emberAfRf4ceZrcHaFillCommandWindowCoveringClusterGoToLiftPercentage(percentageLiftValue)`
- #define `emberAfRf4ceZrcHaFillCommandWindowCoveringClusterGoToTiltValue(tiltValue)`
- #define `emberAfRf4ceZrcHaFillCommandWindowCoveringClusterGoToTiltPercentage(percentageTiltValue)`

Thermostat Commands

- #define `emberAfRf4ceZrcHaFillCommandThermostatClusterSetpointRaiseLower(mode,amount)`

Color Control Commands

- #define `emberAfRf4ceZrcHaFillCommandColorControlClusterMoveToHue(hue,direction,transitionTime)`
- #define `emberAfRf4ceZrcHaFillCommandColorControlClusterMoveHue(moveMode,rate)`
- #define `emberAfRf4ceZrcHaFillCommandColorControlClusterStepHue(stepMode,stepSize,transitionTime)`
- #define `emberAfRf4ceZrcHaFillCommandColorControlClusterMoveToSaturation(saturation,transitionTime)`
- #define `emberAfRf4ceZrcHaFillCommandColorControlClusterMoveSaturation(moveMode, rate)`
- #define `emberAfRf4ceZrcHaFillCommandColorControlClusterStepSaturation(stepMode,stepSize,transitionTime)`
- #define `emberAfRf4ceZrcHaFillCommandColorControlClusterMoveToHueAndSaturation(hue,saturation,transitionTime)`
- #define `emberAfRf4ceZrcHaFillCommandColorControlClusterMoveToColor(colorX,colorY,transitionTime)`
- #define `emberAfRf4ceZrcHaFillCommandColorControlClusterMoveColor(rateX,rateY)`
- #define `emberAfRf4ceZrcHaFillCommandColorControlClusterStepColor(stepX,stepY,transitionTime)`
- #define `emberAfRf4ceZrcHaFillCommandColorControlClusterMoveToColorTemperature(colorTemperature, transitionTime)`

IAS ACE Commands

- #define `emberAfRf4ceZrcHaFillCommandIASACEClusterArm(armMode)`
- #define `emberAfRf4ceZrcHaFillCommandIASACEClusterBypass(numberOfZones,zoneIds,zoneIdsLen)`
- #define `emberAfRf4ceZrcHaFillCommandIASACEClusterEmergency()`
- #define `emberAfRf4ceZrcHaFillCommandIASACEClusterFire()`
- #define `emberAfRf4ceZrcHaFillCommandIASACEClusterPanic()`

8.65 rf4ce-zrc20-ha-client.h

```

00001 // Copyright 2014 Silicon Laboratories, Inc.
00002
00003 #ifndef __RF4CE_ZRC20_HA_CLIENT_H__
00004 #define __RF4CE_ZRC20_HA_CLIENT_H__
00005
00044 typedef enum {
00045     EMBER_ZCL_THERMOSTAT_SETPOINT_RAISE_LOWER_MODE_HEAT
00046         = 0x00,
00047     EMBER_ZCL_THERMOSTAT_SETPOINT_RAISE_LOWER_MODE_COOL
00048         = 0x01,
00049     EMBER_ZCL_THERMOSTAT_SETPOINT_RAISE_LOWER_MODE_BOTH
00050         = 0x02
00048 } EmberAfThermostatSetpointRaiseLowerMode
00049 ;
00049
00050
00051 uint16_t emAfPluginRf4ceZrc20HaFillExternalBuffer
00052     (PGM_P format, ...);
00053 #define emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene (scene)
00054     \
00055             emAfPluginRf4ceZrc20HaFillExternalBuffer("u", \
00056                 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_SCENES_STORE_LOCAL_SCENE_ ## scene)
00057 #define emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene (scene)
00058     \
00059             emAfPluginRf4ceZrc20HaFillExternalBuffer("u", \
00060                 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_SCENES_RECALL_LOCAL_SCENE_ ## scene)
00061
00062 /* Public API. */
00063
00064
00076 EmberStatus emberAfRf4ceZrcHaSend(uint8_t pairingIndex,
00077                                         uint8_t haInstanceId);
00078
00079
00081 // @{
00087 #define emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene0 ()
00088     \
00089             emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene(0)
00090
00096 #define emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene1 ()
00097     \
00098             emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene(1)
00099
00105 #define emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene2 ()
00106     \
00107             emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene(2)
00108
00114 #define emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene3 ()
00115     \
00116             emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene(3)
00117
00123 #define emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene4 ()
00124     \
00125             emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene(4)
00126
00132 #define emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene5 ()
00133     \
00134             emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene(5)
00135
00141 #define emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene6 ()
00142     \
00143             emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene(6)
00144
00150 #define emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene7 ()
00151     \

```

```

00151         emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene(7)
00152
00153
00159 #define emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene8()
00160 \
00160     emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene(8)
00161
00162
00168 #define emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene9()
00169 \
00169     emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene(9)
00170
00171
00177 #define emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene10()
00178 \
00178     emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene(10)
00179
00180
00186 #define emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene11()
00187 \
00187     emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene(11)
00188
00189
00195 #define emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene12()
00196 \
00196     emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene(12)
00197
00198
00204 #define emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene13()
00205 \
00205     emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene(13)
00206
00207
00213 #define emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene14()
00214 \
00214     emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene(14)
00215
00216
00222 #define emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene15()
00223 \
00223     emberAfRf4ceZrcHaFillCommandScenesClusterStoreLocalScene(15)
00224
00225
00231 #define emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene0()
00232 \
00232     emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene(0)
00233
00234
00240 #define emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene1()
00241 \
00241     emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene(1)
00242
00243
00249 #define emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene2()
00250 \
00250     emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene(2)
00251
00252
00258 #define emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene3()
00259 \
00259     emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene(3)
00260
00261
00267 #define emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene4()
00268 \
00268     emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene(4)
00269
00270
00276 #define emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene5()
00277 \
00277     emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene(5)
00278
00279
00285 #define emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene6()
00286 \
00286     emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene(6)
00287
00288
00294 #define emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene7()
00295 \

```

```

00295         emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene(7)
00296
00297
00303 #define emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene8()
00304 \
00305         emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene(8)
00306
00312 #define emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene9()
00313 \
00314         emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene(9)
00315
00321 #define emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene10()
00322 \
00323         emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene(10)
00324
00330 #define emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene11()
00331 \
00332         emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene(11)
00333
00339 #define emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene12()
00340 \
00341         emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene(12)
00342
00348 #define emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene13()
00349 \
00350         emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene(13)
00351
00357 #define emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene14()
00358 \
00359         emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene(14)
00360
00366 #define emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene15()
00367 \
00368         emberAfRf4ceZrcHaFillCommandScenesClusterRecallLocalScene(15)
00369
00372 // @{
00378 #define emberAfRf4ceZrcHaFillCommandOnOffClusterOff()
00379 \
00380         emAfPluginRf4ceZrc20HaFillExternalBuffer("u",
00381 \
00382             EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_ON_OFF_OFF)
00383
00388 #define emberAfRf4ceZrcHaFillCommandOnOffClusterOn()
00389 \
00390         emAfPluginRf4ceZrc20HaFillExternalBuffer("u",
00391 \
00392             EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_ON_OFF_ON)
00393
00398 #define emberAfRf4ceZrcHaFillCommandOnOffClusterToggle()
00399 \
00400         emAfPluginRf4ceZrc20HaFillExternalBuffer("u",
00401 \
00402             EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_ON_OFF_TOGGLE)
00403
00405 // @{
00413 #define emberAfRf4ceZrcHaFillCommandLevelControlClusterMoveToLevel(level,
00414 \
00415     transitionTime) \
00416         emAfPluginRf4ceZrc20HaFillExternalBuffer("uuv",
00417 \
00418             EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_LEVEL_CONTROL_MOVE_TO_LEVEL, \
00419                 level,
00420 \
00421                 transitionTime)
00428 #define emberAfRf4ceZrcHaFillCommandLevelControlClusterMove(moveMode,
00429 \
00430             rate)

```

```

00430 \
00431     emAfPluginRf4ceZrc20HaFillExternalBuffer("uuu",
00432         EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_LEVEL_CONTROL_MOVE, \
00433             moveMode,
00434                 \
00435                     rate)
00436
00437 #define emberAfRf4ceZrcHaFillCommandLevelControlClusterStep(stepMode,
00438 \
00439             \
00440                 stepSize,
00441 \
00442             transitionTime)
00443 \
00444     emAfPluginRf4ceZrc20HaFillExternalBuffer("uuuv",
00445         EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_LEVEL_CONTROL_STEP, \
00446             stepMode,
00447                 \
00448                     stepSize,
00449                         \
00450                             transitionTime)
00451
00452
00453 #define emberAfRf4ceZrcHaFillCommandLevelControlClusterStop()
00454 \
00455     emAfPluginRf4ceZrc20HaFillExternalBuffer("u",
00456         EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_LEVEL_CONTROL_STOP)
00457
00458
00459 #define emberAfRf4ceZrcHaFillCommandLevelControlClusterMoveToLevelWithOnOff(level,
00460 \
00461             transitionTime) \
00462     emAfPluginRf4ceZrc20HaFillExternalBuffer("uuv",
00463             \
00464     EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_LEVEL_CONTROL_MOVE_TO_LEVEL_WITH_ON_OFF, \
00465         level,
00466             \
00467                 transitionTime)
00468
00469
00470 #define emberAfRf4ceZrcHaFillCommandLevelControlClusterMoveWithOnOff(moveMode,
00471 \
00472             \
00473                 rate)
00474 \
00475     emAfPluginRf4ceZrc20HaFillExternalBuffer("uuu",
00476         EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_LEVEL_CONTROL_MOVE_WITH_ON_OFF, \
00477             moveMode,
00478                 \
00479                     rate)
00480
00481
00482 #define emberAfRf4ceZrcHaFillCommandLevelControlClusterStepWithOnOff(stepMode,
00483 \
00484             \
00485                 stepSize,
00486 \
00487             transitionTime) \
00488     emAfPluginRf4ceZrc20HaFillExternalBuffer("uuuv",
00489             \
00490     EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_LEVEL_CONTROL_STEP_WITH_ON_OFF, \
00491         stepMode,
00492             \
00493                 stepSize,
00494                     \
00495                         transitionTime)
00496
00497
00498 #define emberAfRf4ceZrcHaFillCommandLevelControlClusterStopWithOnOff()
00499 \
00500     emAfPluginRf4ceZrc20HaFillExternalBuffer("u",
00501         EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_LEVEL_CONTROL_STOP_WITH_ON_OFF)
00502

```

```

00524 // @@
00531 #define emberAfRf4ceZrcHaFillCommandDoorLockClusterLockDoor(pinRfidCode)
\           emAfPluginRf4ceZrc20HaFillExternalBuffer("us",
00532           EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_DOOR_LOCK_LOCK_DOOR, \
00533                           pinRfidCode)
00534
00535
00536
00543 #define emberAfRf4ceZrcHaFillCommandDoorLockClusterUnlockDoor(pinRfidCode)
\           emAfPluginRf4ceZrc20HaFillExternalBuffer("us",
00544           EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_DOOR_LOCK_UNLOCK_DOOR, \
00545                           pinRfidCode)
00546
00547
00548
00555 #define emberAfRf4ceZrcHaFillCommandDoorLockClusterToggle(pinRfidCode)
\           emAfPluginRf4ceZrc20HaFillExternalBuffer("us",
00556           EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_DOOR_LOCK_TOGGLE, \
00557                           pinRfidCode)
00558
00559
00560
00568 #define
emberAfRf4ceZrcHaFillCommandDoorLockClusterUnlockWithTimeout(timeoutInSeconds, \
00569   pinRfidCode) \
00570   emAfPluginRf4ceZrc20HaFillExternalBuffer("uvs",
\           \
00571   EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_DOOR_LOCK_TOGGLE,
00572                           timeoutInSeconds,
\           \
00573                           pinRfidCode)
00574
00578 // @@
00584 #define emberAfRf4ceZrcHaFillCommandWindowCoveringClusterUpOpen()
\           emAfPluginRf4ceZrc20HaFillExternalBuffer("u",
00585           EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_WINDOW_COVERING_UP_OPEN)
00586
00587
00588
00594 #define emberAfRf4ceZrcHaFillCommandWindowCoveringClusterDownClose()
\           emAfPluginRf4ceZrc20HaFillExternalBuffer("u",
00595           EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_WINDOW_COVERING_DOWN_CLOSE)
00596
00597
00598
00604 #define emberAfRf4ceZrcHaFillCommandWindowCoveringClusterStop()
\           emAfPluginRf4ceZrc20HaFillExternalBuffer("u",
00605           EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_WINDOW_COVERING_STOP)
00606
00607
00608
00615 #define
emberAfRf4ceZrcHaFillCommandWindowCoveringClusterGoToLiftValue(liftValue) \
00616   emAfPluginRf4ceZrc20HaFillExternalBuffer("uv",
\           \
00617   EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_WINDOW_COVERING_GO_TO_LIFT_VALUE,
00618                           liftValue)
00619
00620
00627 #define
emberAfRf4ceZrcHaFillCommandWindowCoveringClusterGoToLiftPercentage(percentageLiftValue) \
00628   emAfPluginRf4ceZrc20HaFillExternalBuffer("uu",
\           \
00629   EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_WINDOW_COVERING_GO_TO_LIFT_PERCENTAGE,
00630                           percentageLiftValue)
00631
00632
00639 #define
emberAfRf4ceZrcHaFillCommandWindowCoveringClusterGoToTiltValue(tiltValue) \
00640   emAfPluginRf4ceZrc20HaFillExternalBuffer("uv",
\           \

```

```

00641 \
00642     EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_WINDOW_COVERING_GO_TO_TILT_VALUE,      \
00643                                         tiltValue)
00644
00645 #define
00646     emberAfRf4ceZrcHaFillCommandWindowCoveringClusterGoToTiltPercentage(percentageTiltValue)      \
00647         emAfPluginRf4ceZrc20HaFillExternalBuffer("uu",
00648             \
00649             EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_WINDOW_COVERING_GO_TO_TILT_PERCENTAGE,      \
00650                                         percentageTiltValue)
00651 // @{
00652 #define emberAfRf4ceZrcHaFillCommandThermostatClusterSetpointRaiseLower(mode,
00653 \
00654             \
00655             EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_THERMOSTAT_SETPOINT_RAISE_LOWER, \
00656                                         mode,
00657             \
00658             amount)
00659 // @{
00660 #define emberAfRf4ceZrcHaFillCommandColorControlClusterMoveToHue(hue,
00661 \
00662             \
00663             EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_COLOR_CONTROL_MOVE_TO_HUE, \
00664                                         hue,
00665             \
00666             direction,
00667             \
00668             transitionTime)\ \
00669                 emAfPluginRf4ceZrc20HaFillExternalBuffer("uuuv", \
00670                     \
00671                     EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_COLOR_CONTROL_MOVE_TO_HUE,      \
00672                                         hue,
00673             \
00674             direction,
00675             \
00676             transitionTime)
00677 // @{
00678 #define emberAfRf4ceZrcHaFillCommandColorControlClusterMoveHue(moveMode,
00679 \
00680             \
00681             EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_COLOR_CONTROL_MOVE_HUE, \
00682                                         moveMode,
00683             \
00684             rate)
00685 \
00686             emAfPluginRf4ceZrc20HaFillExternalBuffer("uuu", \
00687                 \
00688                 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_COLOR_CONTROL_MOVE_HUE,      \
00689                                         moveMode,
00690             \
00691             rate)
00692
00693
00694
00695
00696 #define emberAfRf4ceZrcHaFillCommandColorControlClusterStepHue(stepMode,
00697 \
00698             \
00699             EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_COLOR_CONTROL_STEP_HUE, \
00700                                         stepMode,
00701             \
00702             stepSize,
00703             \
00704             transitionTime)
00705 \
00706             emAfPluginRf4ceZrc20HaFillExternalBuffer("uuuu", \
00707                 \
00708                 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_COLOR_CONTROL_STEP_HUE,      \
00709                                         stepMode,
00710             \
00711             stepSize,
00712             \
00713             transitionTime)
00714
00715
00716
00717
00718
00719 #define emberAfRf4ceZrcHaFillCommandColorControlClusterMoveToSaturation(saturation,
00720 \
00721             \
00722             EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_COLOR_CONTROL_MOVE_TO_SATURATION, \
00723                                         saturation,
00724             \
00725             transitionTime)\ \
00726                 emAfPluginRf4ceZrc20HaFillExternalBuffer("uuv",
00727             \
00728             EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_COLOR_CONTROL_MOVE_TO_SATURATION,      \
00729                                         saturation,
00730             \
00731             transitionTime)
00732
00733
00734
00735
00736
00737
00738
00739
00740

```

```

00741                                     \
00742                                     transitionTime)
00743
00751 #define emberAfRf4ceZrcHaFillCommandColorControlClusterMoveSaturation(moveMode,
00752   \
00752   \
00753     emAfPluginRf4ceZrc20HaFillExternalBuffer("uuu",                                \
00754                                         EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_COLOR_CONTROL_MOVE_SATURATION, \
00755                                         moveMode,                                         \
00756                                         rate)
00757
00758
00767 #define emberAfRf4ceZrcHaFillCommandColorControlClusterStepSaturation(stepMode,
00768   \
00768   \
00769     transitionTime) \
00770       emAfPluginRf4ceZrc20HaFillExternalBuffer("uuuu", \
00771                                         EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_COLOR_CONTROL_STEP_SATURATION, \
00772                                         stepMode, \
00773                                         stepSize, \
00774                                         transitionTime)
00775
00776
00785 #define
00786   emberAfRf4ceZrcHaFillCommandColorControlClusterMoveToHueAndSaturation(hue, \
00787   saturation, \
00788   transitionTime) \
00789     emAfPluginRf4ceZrc20HaFillExternalBuffer("uuuv", \
00790                                         EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_COLOR_CONTROL_MOVE_TO_HUE_AND_SATURATION, \
00791                                         hue, \
00792                                         saturation, \
00793                                         transitionTime)
00794
00803 #define emberAfRf4ceZrcHaFillCommandColorControlClusterMoveToColor(colorX,
00804   \
00804   \
00805   transitionTime) \
00806     emAfPluginRf4ceZrc20HaFillExternalBuffer("uvvv", \
00807                                         EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_COLOR_CONTROL_MOVE_TO_COLOR, \
00808                                         colorX, \
00809                                         colorY, \
00810                                         transitionTime)
00811
00812
00820 #define emberAfRf4ceZrcHaFillCommandColorControlClusterMoveColor(rateX,
00821   \
00821   \
00822     emAfPluginRf4ceZrc20HaFillExternalBuffer("uvv", \
00823                                         EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_COLOR_CONTROL_MOVE_COLOR, \
00824                                         rateX, \
00825                                         rateY)
00826
00827
00836 #define emberAfRf4ceZrcHaFillCommandColorControlClusterStepColor(stepX,

```

```

00837 \
00838 \
00839     transitionTime) \
00840         emAfPluginRf4ceZrc20HaFillExternalBuffer("uvvv", \
00841             EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_COLOR_CONTROL_STEP_COLOR, \
00842                 stepX, \
00843                     stepY, \
00844                         transitionTime)
00845
00846
00847 #define
00848     emberAfRf4ceZrcHaFillCommandColorControlClusterMoveToColorTemperature(colorTemperature, \
00849         transitionTime) \
00850             emAfPluginRf4ceZrc20HaFillExternalBuffer("uvv", \
00851                 \
00852                 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_MOVE_TO_COLOR_TEMPERATURE, \
00853                     colorTemperature, \
00854                         transitionTime)
00855
00856
00857 // @{
00858 #define emberAfRf4ceZrcHaFillCommandIASACEClusterArm(armMode)
00859 \
00860         emAfPluginRf4ceZrc20HaFillExternalBuffer("uu", \
00861             EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_IAS_ACE_ARM, \
00862                 armMode)
00863
00864 #define emberAfRf4ceZrcHaFillCommandIASACEClusterBypass(numberOfZones,
00865 \
00866             zoneIds,
00867                 zoneIdsLen)
00868 \
00869         emAfPluginRf4ceZrc20HaFillExternalBuffer("uub", \
00870             EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_IAS_ACE_BYPASS, \
00871                 numberOfZones, \
00872                     zoneIds,
00873                         zoneIdsLen)
00874
00875
00876 #define emberAfRf4ceZrcHaFillCommandIASACEClusterEmergency()
00877 \
00878         emAfPluginRf4ceZrc20HaFillExternalBuffer("u",
00879             EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_IAS_ACE_EMERGENCY)
00880
00881
00882 #define emberAfRf4ceZrcHaFillCommandIASACEClusterFire()
00883 \
00884         emAfPluginRf4ceZrc20HaFillExternalBuffer("u",
00885             EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_IAS_ACE_FIRE)
00886
00887
00888 #define emberAfRf4ceZrcHaFillCommandIASACEClusterPanic()
00889 \
00890         emAfPluginRf4ceZrc20HaFillExternalBuffer("u",
00891             EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_IAS_ACE_PANIC)
00892
00893 #define emberAfRf4ceZrcHaFillCommandPreviousDestinationGroup()
00894 \
00895         emAfPluginRf4ceZrc20HaFillExternalBuffer("u",
00896             EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_PREVIOUS_DESTINATION_GROUP)
00897
00898 #define emberAfRf4ceZrcHaFillCommandNextDestinationGroup()
00899

```

```

00939 \
00940     emAfPluginRf4ceZrc20HaFillExternalBuffer("u",
00941     EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_NEXT_DESTINATION_GROUP)
00942 #endif // __RF4CE_ZRC20_HA_CLIENT_H__
00943
00944 // END addtogroup

```

8.66 rf4ce-zrc20-ha-server-tokens.h File Reference

Macros

- `#define CREATOR_PLUGIN_RF4CE_ZRC20_HA_SERVER_LOGICAL_DEVICES_TABLE`
- `#define CREATOR_PLUGIN_RF4CE_ZRC20_HA_SERVER_INSTANCE_TO_LOGICAL_DEVICE_TABLE`

8.66.1 Macro Definition Documentation

8.66.1.1 `#define CREATOR_PLUGIN_RF4CE_ZRC20_HA_SERVER_LOGICAL_DEVICES_TABLE`

Definition at line 4 of file [rf4ce-zrc20-ha-server-tokens.h](#).

8.66.1.2 `#define CREATOR_PLUGIN_RF4CE_ZRC20_HA_SERVER_INSTANCE_TO_LOGICAL_DEVICE_TABLE`

Definition at line 5 of file [rf4ce-zrc20-ha-server-tokens.h](#).

8.67 rf4ce-zrc20-ha-server-tokens.h

```

00001 // Copyright 2014 Silicon Laboratories, Inc.
00002
00003
00004 #define CREATOR_PLUGIN_RF4CE_ZRC20_HA_SERVER_LOGICAL_DEVICES_TABLE
0x8735
00005 #define CREATOR_PLUGIN_RF4CE_ZRC20_HA_SERVER_INSTANCE_TO_LOGICAL_DEVICE_TABLE
0x8736
00006
00007 #ifdef DEFINETYPES
00008 #include "app/framework/plugin/rf4ce-profile/rf4ce-profile-types.h"
00009 #include "app/framework/plugin/rf4ce-zrc20/rf4ce-zrc20-types.h"
00010 #include "rf4ce-zrc20-ha-server.h"
00011
00012 typedef struct {
00013     uint8_t instances[ZRC_HA_SERVER_NUM_OF_HA_INSTANCES];
00014 } InstStruct;
00015 #endif
00016
00017
00018 #ifdef DEFINETOKENS
00019 DEFINE_INDEXED_TOKEN(PLUGIN_RF4CE_ZRC20_HA_SERVER_LOGICAL_DEVICES_TABLE,
00020                         DestStruct,
00021                         (
00022                             EMBER_AF_PLUGIN_RF4CE_ZRC20_HA_SERVER_LOGICAL_DEVICES_TABLE_SIZE),
00023                             {0,})
00024 DEFINE_INDEXED_TOKEN(
00025     PLUGIN_RF4CE_ZRC20_HA_SERVER_INSTANCE_TO_LOGICAL_DEVICE_TABLE,
00026             InstStruct,
00027             (EMBER_RF4CE_PAIRING_TABLE_SIZE),

```

```

00027             {{0,}})
00028 #endif
00029

```

8.68 rf4ce-zrc20-ha-server.h File Reference

```
#include "app/framework/plugin/rf4ce-gdp/rf4ce-gdp-types.h"
```

Data Structures

- struct DestStruct
- struct HaAttributesInfo

Macros

- #define EMBER_AF_PLUGIN_RF4CE_ZRC20_HA_SERVER_ZCL_BUFFER_SIZE
- #define ZRC_HA_SERVER_NUM_OF_HA_INSTANCES
- #define ZRC_ACTION_ID_HIGH_NIBBLE_MASK
- #define ZRC_ACTION_ID_LOW_NIBBLE_MASK
- #define ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE0_ID
- #define ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE1_ID
- #define ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE2_ID
- #define ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE3_ID
- #define ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE4_ID
- #define ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE5_ID
- #define ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE6_ID
- #define ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE7_ID
- #define ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE8_ID
- #define ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE9_ID
- #define ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE10_ID
- #define ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE11_ID
- #define ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE12_ID
- #define ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE13_ID
- #define ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE14_ID
- #define ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE15_ID
- #define ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE_ID_SIZE
- #define ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE_SIZE
- #define ZRC_HA_SCENE_LOCAL_SCENE_VALID0_ID
- #define ZRC_HA_SCENE_LOCAL_SCENE_VALID1_ID
- #define ZRC_HA_SCENE_LOCAL_SCENE_VALID2_ID
- #define ZRC_HA_SCENE_LOCAL_SCENE_VALID3_ID
- #define ZRC_HA_SCENE_LOCAL_SCENE_VALID4_ID
- #define ZRC_HA_SCENE_LOCAL_SCENE_VALID5_ID
- #define ZRC_HA_SCENE_LOCAL_SCENE_VALID6_ID
- #define ZRC_HA_SCENE_LOCAL_SCENE_VALID7_ID
- #define ZRC_HA_SCENE_LOCAL_SCENE_VALID8_ID
- #define ZRC_HA_SCENE_LOCAL_SCENE_VALID9_ID
- #define ZRC_HA_SCENE_LOCAL_SCENE_VALID10_ID

- #define ZRC_HA_SCENE_LOCAL_SCENE_VALID11_ID
- #define ZRC_HA_SCENE_LOCAL_SCENE_VALID12_ID
- #define ZRC_HA_SCENE_LOCAL_SCENE_VALID13_ID
- #define ZRC_HA_SCENE_LOCAL_SCENE_VALID14_ID
- #define ZRC_HA_SCENE_LOCAL_SCENE_VALID15_ID
- #define ZRC_HA_SCENE_LOCAL_SCENE_VALID_ID_SIZE
- #define ZRC_HA_SCENE_LOCAL_SCENE_VALID_SIZE
- #define ZRC_HA_ON_OFF_ON_OFF_ID
- #define ZRC_HA_ON_OFF_ON_OFF_ID_SIZE
- #define ZRC_HA_ON_OFF_ON_OFF_SIZE
- #define ZRC_HA_LEVEL_CONTROL_CURRENT_LEVEL_ID
- #define ZRC_HA_LEVEL_CONTROL_CURRENT_LEVEL_ID_SIZE
- #define ZRC_HA_LEVEL_CONTROL_CURRENT_LEVEL_SIZE
- #define ZRC_HA_DOOR_LOCK_LOCK_DOOR_RESPONSE_ID
- #define ZRC_HA_DOOR_LOCK_LOCK_DOOR_RESPONSE_ID_SIZE
- #define ZRC_HA_DOOR_LOCK_LOCK_DOOR_RESPONSE_SIZE
- #define ZRC_HA_DOOR_LOCK_UNLOCK_DOOR_RESPONSE_ID
- #define ZRC_HA_DOOR_LOCK_UNLOCK_DOOR_RESPONSE_ID_SIZE
- #define ZRC_HA_DOOR_LOCK_UNLOCK_DOOR_RESPONSE_SIZE
- #define ZRC_HA_DOOR_LOCK_TOGGLE_RESPONSE_ID
- #define ZRC_HA_DOOR_LOCK_TOGGLE_RESPONSE_ID_SIZE
- #define ZRC_HA_DOOR_LOCK_TOGGLE_RESPONSE_SIZE
- #define ZRC_HA_DOOR_LOCK_UNLOCK_WITH_TIMEOUT_RESPONSE_ID
- #define ZRC_HA_DOOR_LOCK_UNLOCK_WITH_TIMEOUT_RESPONSE_ID_SIZE
- #define ZRC_HA_DOOR_LOCK_UNLOCK_WITH_TIMEOUT_RESPONSE_SIZE
- #define ZRC_HA_WINDOW_COVERING_CURRENT_POSITION_LIFT_PERCENTAGE_ID
- #define ZRC_HA_WINDOW_COVERING_CURRENT_POSITION_LIFT_PERCENTAGE_ID_SIZE
- #define ZRC_HA_WINDOW_COVERING_CURRENT_POSITION_LIFT_PERCENTAGE_SIZE
- #define ZRC_HA_WINDOW_COVERING_CURRENT_POSITION_TILT_PERCENTAGE_ID
- #define ZRC_HA_WINDOW_COVERING_CURRENT_POSITION_TILT_PERCENTAGE_ID_SIZE
- #define ZRC_HA_WINDOW_COVERING_CURRENT_POSITION_TILT_PERCENTAGE_SIZE
- #define ZRC_HA_THERMOSTAT_LOCAL_TEMPERATURE_ID
- #define ZRC_HA_THERMOSTAT_LOCAL_TEMPERATURE_ID_SIZE
- #define ZRC_HA_THERMOSTAT_LOCAL_TEMPERATURE_SIZE
- #define ZRC_HA_THERMOSTAT_OCCUPIED_COOLING_SETPOINT_ID
- #define ZRC_HA_THERMOSTAT_OCCUPIED_COOLING_SETPOINT_ID_SIZE
- #define ZRC_HA_THERMOSTAT_OCCUPIED_COOLING_SETPOINT_SIZE
- #define ZRC_HA_THERMOSTAT_OCCUPIED_HEATING_SETPOINT_ID
- #define ZRC_HA_THERMOSTAT_OCCUPIED_HEATING_SETPOINT_ID_SIZE
- #define ZRC_HA_THERMOSTAT_OCCUPIED_HEATING_SETPOINT_SIZE
- #define ZRC_HA_COLOR_CONTROL_CURRENT_HUE_ID
- #define ZRC_HA_COLOR_CONTROL_CURRENT_HUE_ID_SIZE
- #define ZRC_HA_COLOR_CONTROL_CURRENT_HUE_SIZE
- #define ZRC_HA_COLOR_CONTOL_CURRENT_SATURATION_ID
- #define ZRC_HA_COLOR_CONTOL_CURRENT_SATURATION_ID_SIZE
- #define ZRC_HA_COLOR_CONTOL_CURRENT_SATURATION_SIZE

- #define ZRC_HA_COLOR_CONTOL_CURRENT_X_ID
- #define ZRC_HA_COLOR_CONTOL_CURRENT_X_ID_SIZE
- #define ZRC_HA_COLOR_CONTOL_CURRENT_X_SIZE
- #define ZRC_HA_COLOR_CONTOL_CURRENT_Y_ID
- #define ZRC_HA_COLOR_CONTOL_CURRENT_Y_ID_SIZE
- #define ZRC_HA_COLOR_CONTOL_CURRENT_Y_SIZE
- #define ZRC_HA_COLOR_CONTOL_COLOR_MODE_ID
- #define ZRC_HA_COLOR_CONTOL_COLOR_MODE_ID_SIZE
- #define ZRC_HA_COLOR_CONTOL_COLOR_MODE_SIZE
- #define ZRC_HA_IAS_ACE_ARM_RESPONSE_ID
- #define ZRC_HA_IAS_ACE_ARM_RESPONSE_ID_SIZE
- #define ZRC_HA_IAS_ACE_ARM_RESPONSE_SIZE
- #define ZRC_HA_ATTRIBUTE_STATUS_TABLE_SIZE
- #define ZRC_HA_ATTRIBUTE_TABLE_SIZE
- #define HA_ATTRIBUTE_STATUS_LENGTH
- #define HA_ATTRIBUTE_STATUS_CHANGED_FLAG

Functions

- void emberAfPluginRf4ceZrc20HaServerClearAllHaAttributes (void)
- EmberAfRf4ceGdpAttributeStatus emberAfPluginRf4ceZrc20HaServerGetHaAttribute (uint8_t pairingIndex, uint8_t haInstanceId, uint8_t haAttributeId, EmberAfRf4ceZrcHomeAutomationAttribute *haAttribute)
- EmberAfRf4ceGdpAttributeStatus emberAfPluginRf4ceZrc20HaServerSetHaAttribute (uint8_t pairingIndex, uint8_t haInstanceId, uint8_t haAttributeId, EmberAfRf4ceZrcHomeAutomationAttribute *haAttribute)
- EmberStatus emberAfPluginRf4ceZrc20HaMappingToLogicalDeviceAdd (uint8_t pairingIndex, uint8_t haInstanceId, uint8_t destIndex)
- EmberStatus emberAfPluginRf4ceZrc20HaMappingToLogicalDeviceRemove (uint8_t destIndex)
- EmberStatus emberAfPluginRf4ceZrc20HaMappingToLogicalDeviceGet (uint8_t pairingIndex, uint8_t haInstanceId, uint8_t *destIndex)
- EmberStatus emberAfPluginRf4ceZrc20HaLogicalDeviceDestinationAdd (DestStruct *dest, uint8_t *index)
- EmberStatus emberAfPluginRf4ceZrc20HaLogicalDeviceDestinationRemove (DestStruct *dest, uint8_t *index)
- uint8_t emberAfPluginRf4ceZrc20HaLogicalDeviceDestinationTableSize (void)
- EmberStatus emberAfPluginRf4ceZrc20HaLogicalDeviceDestinationGet (uint8_t pairingIndex, uint8_t haInstanceId, DestStruct *dest)
- EmberStatus emberAfPluginRf4ceZrc20HaLogicalDeviceIndexLookUp (DestStruct *dest, uint8_t *index)
- void emberAfPluginRf4ceZrc20HaLogicalDeviceAndInstanceToLogicalDeviceMappingClear (void)
- EmberStatus emberAfPluginRf4ceZrc20HaLogicalDeviceAndInstanceToLogicalDeviceMappingAdd (uint8_t pairingIndex, uint8_t haInstanceId, DestStruct *dest)
- EmberStatus emberAfPluginRf4ceZrc20HaLogicalDeviceAndInstanceToLogicalDeviceMappingRemove (DestStruct *dest)
- EmberStatus emAfRf4ceZrc20ParseHaActionAndForwardToZclNetwork (const EmberAfRf4ceZrcActionRecord *record)
- void emAfRf4ceZrc20ClearLogicalDevicesTable (void)
- void emAfRf4ceZrc20ClearInstanceToLogicalDeviceTable (void)
- EmberStatus emAfRf4ceZrc20AddLogicalDeviceDestination (DestStruct *dest, uint8_t *index)
- EmberStatus emAfRf4ceZrc20RemoveLogicalDeviceDestination (uint8_t destIndex)
- uint8_t GetLogicalDeviceDestination (uint8_t i, DestStruct *dest)
- void DestLookup (uint8_t pairingIndex, uint8_t haInstanceId, DestStruct *dest)

8.69 rf4ce-zrc20-ha-server.h

```

00001 // Copyright 2014 Silicon Laboratories, Inc.
00002
00003 #ifndef __RF4CE_ZRC20_HA_SERVER_H__
00004 #define __RF4CE_ZRC20_HA_SERVER_H__
00005
00006 #include "app/framework/plugin/rf4ce-gdp/rf4ce-gdp-types.h"
00007
00042 // EMBER_AF_MAXIMUM_SEND_PAYLOAD_LENGTH is set in config.h
00043 #define EMBER_AF_PLUGIN_RF4CE_ZRC20_HA_SERVER_ZCL_BUFFER_SIZE
    EMBER_AF_MAXIMUM_SEND_PAYLOAD_LENGTH
00044
00045 #define ZRC_HA_SERVER_NUM_OF_HA_INSTANCES      32
00046
00047 #define ZRC_ACTION_ID_HIGH_NIBBLE_MASK          0xFO
00048 #define ZRC_ACTION_ID_LOW_NIBBLE_MASK           0xOF
00049
00050
00051 /* ZRC HA Attribute IDs and size.
00052 * Size contains HA attribute status and HA attribute value. */
00053 #define ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE0_ID      0x00
00054 #define ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE1_ID      0x01
00055 #define ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE2_ID      0x02
00056 #define ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE3_ID      0x03
00057 #define ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE4_ID      0x04
00058 #define ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE5_ID      0x05
00059 #define ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE6_ID      0x06
00060 #define ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE7_ID      0x07
00061 #define ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE8_ID      0x08
00062 #define ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE9_ID      0x09
00063 #define ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE10_ID     0x0A
00064 #define ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE11_ID     0x0B
00065 #define ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE12_ID     0x0C
00066 #define ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE13_ID     0x0D
00067 #define ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE14_ID     0x0E
00068 #define ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE15_ID     0x0F
00069 #define ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE_ID_SIZE   1
00070 #define ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE_SIZE       1
00071 #define ZRC_HA_SCENE_LOCAL_SCENE_VALID0_ID                0x10
00072 #define ZRC_HA_SCENE_LOCAL_SCENE_VALID1_ID                0x11
00073 #define ZRC_HA_SCENE_LOCAL_SCENE_VALID2_ID                0x12
00074 #define ZRC_HA_SCENE_LOCAL_SCENE_VALID3_ID                0x13
00075 #define ZRC_HA_SCENE_LOCAL_SCENE_VALID4_ID                0x14
00076 #define ZRC_HA_SCENE_LOCAL_SCENE_VALID5_ID                0x15
00077 #define ZRC_HA_SCENE_LOCAL_SCENE_VALID6_ID                0x16
00078 #define ZRC_HA_SCENE_LOCAL_SCENE_VALID7_ID                0x17
00079 #define ZRC_HA_SCENE_LOCAL_SCENE_VALID8_ID                0x18
00080 #define ZRC_HA_SCENE_LOCAL_SCENE_VALID9_ID                0x19
00081 #define ZRC_HA_SCENE_LOCAL_SCENE_VALID10_ID               0x1A
00082 #define ZRC_HA_SCENE_LOCAL_SCENE_VALID11_ID               0x1B
00083 #define ZRC_HA_SCENE_LOCAL_SCENE_VALID12_ID               0x1C
00084 #define ZRC_HA_SCENE_LOCAL_SCENE_VALID13_ID               0x1D
00085 #define ZRC_HA_SCENE_LOCAL_SCENE_VALID14_ID               0x1E
00086 #define ZRC_HA_SCENE_LOCAL_SCENE_VALID15_ID               0x1F
00087 #define ZRC_HA_SCENE_LOCAL_SCENE_VALID_ID_SIZE            1
00088 #define ZRC_HA_SCENE_LOCAL_SCENE_VALID_SIZE              1
00089 #define ZRC_HA_ON_OFF_ON_OFF_ID                         0x20
00090 #define ZRC_HA_ON_OFF_ON_OFF_ID_SIZE                   1
00091 #define ZRC_HA_ON_OFF_ON_SIZE                          1
00092 #define ZRC_HA_LEVEL_CONTROL_CURRENT_LEVEL_ID          0x30
00093 #define ZRC_HA_LEVEL_CONTROL_CURRENT_LEVEL_ID_SIZE     1
00094 #define ZRC_HA_LEVEL_CONTROL_CURRENT_LEVEL_SIZE        1
00095 #define ZRC_HA_DOOR_LOCK_LOCK_DOOR_RESPONSE_ID         0x40
00096 #define ZRC_HA_DOOR_LOCK_LOCK_DOOR_RESPONSE_ID_SIZE    1
00097 #define ZRC_HA_DOOR_LOCK_LOCK_DOOR_RESPONSE_SIZE       1
00098 #define ZRC_HA_DOOR_LOCK_UNLOCK_DOOR_RESPONSE_ID        0x41
00099 #define ZRC_HA_DOOR_LOCK_UNLOCK_DOOR_RESPONSE_ID_SIZE  1
00100 #define ZRC_HA_DOOR_LOCK_UNLOCK_DOOR_RESPONSE_SIZE     1
00101 #define ZRC_HA_DOOR_LOCK_TOGGLE_RESPONSE_ID             0x42
00102 #define ZRC_HA_DOOR_LOCK_TOGGLE_RESPONSE_ID_SIZE       1
00103 #define ZRC_HA_DOOR_LOCK_TOGGLE_RESPONSE_SIZE          1
00104 #define ZRC_HA_DOOR_LOCK_UNLOCK_WITH_TIMEOUT_RESPONSE_ID 0x43
00105 #define ZRC_HA_DOOR_LOCK_UNLOCK_WITH_TIMEOUT_RESPONSE_ID_SIZE 1
00106 #define ZRC_HA_DOOR_LOCK_UNLOCK_WITH_TIMEOUT_RESPONSE_SIZE 1
00107 #define ZRC_HA_WINDOW_COVERING_CURRENT_POSITION_LIFT_PERCENTAGE_ID 0x50
00108 #define ZRC_HA_WINDOW_COVERING_CURRENT_POSITION_LIFT_PERCENTAGE_ID_SIZE 1
00109 #define ZRC_HA_WINDOW_COVERING_CURRENT_POSITION_LIFT_PERCENTAGE_SIZE 1

```

```

00110 #define ZRC_HA_WINDOW_COVERING_CURRENT_POSITION_TILT_PERCENTAGE_ID      0x51
00111 #define ZRC_HA_WINDOW_COVERING_CURRENT_POSITION_TILT_PERCENTAGE_ID_SIZE  1
00112 #define ZRC_HA_WINDOW_COVERING_CURRENT_POSITION_TILT_PERCENTAGE_SIZE    1
00113 #define ZRC_HA_THERMOSTAT_LOCAL_TEMPERATURE_ID                         0x60
00114 #define ZRC_HA_THERMOSTAT_LOCAL_TEMPERATURE_ID_SIZE                     1
00115 #define ZRC_HA_THERMOSTAT_LOCAL_TEMPERATURE_SIZE                      2
00116 #define ZRC_HA_THERMOSTAT_OCCUPIED_COOLING_SETPOINT_ID                 0x61
00117 #define ZRC_HA_THERMOSTAT_OCCUPIED_COOLING_SETPOINT_ID_SIZE           1
00118 #define ZRC_HA_THERMOSTAT_OCCUPIED_COOLING_SETPOINT_SIZE               2
00119 #define ZRC_HA_THERMOSTAT_OCCUPIED_HEATING_SETPOINT_ID                  0x62
00120 #define ZRC_HA_THERMOSTAT_OCCUPIED_HEATING_SETPOINT_ID_SIZE             1
00121 #define ZRC_HA_THERMOSTAT_OCCUPIED_HEATING_SETPOINT_SIZE                2
00122 #define ZRC_HA_COLOR_CONTROL_CURRENT_HUE_ID                           0x70
00123 #define ZRC_HA_COLOR_CONTROL_CURRENT_HUE_ID_SIZE                     1
00124 #define ZRC_HA_COLOR_CONTROL_CURRENT_HUE_SIZE                       1
00125 #define ZRC_HA_COLOR_CONTOL_CURRENT_SATURATION_ID                   0x71
00126 #define ZRC_HA_COLOR_CONTOL_CURRENT_SATURATION_ID_SIZE             1
00127 #define ZRC_HA_COLOR_CONTOL_CURRENT_SATURATION_SIZE                1
00128 #define ZRC_HA_COLOR_CONTOL_CURRENT_X_ID                            0x72
00129 #define ZRC_HA_COLOR_CONTOL_CURRENT_X_ID_SIZE                     1
00130 #define ZRC_HA_COLOR_CONTOL_CURRENT_X_SIZE                       2
00131 #define ZRC_HA_COLOR_CONTOL_CURRENT_Y_ID                           0x73
00132 #define ZRC_HA_COLOR_CONTOL_CURRENT_Y_ID_SIZE                     1
00133 #define ZRC_HA_COLOR_CONTOL_CURRENT_Y_SIZE                       2
00134 #define ZRC_HA_COLOR_CONTOL_COLOR_MODE_ID                         0x74
00135 #define ZRC_HA_COLOR_CONTOL_COLOR_MODE_ID_SIZE                   1
00136 #define ZRC_HA_COLOR_CONTOL_COLOR_MODE_SIZE                     1
00137 #define ZRC_HA_IAS_ACE_ARM_RESPONSE_ID                          0xC4
00138 #define ZRC_HA_IAS_ACE_ARM_RESPONSE_ID_SIZE                   1
00139 #define ZRC_HA_IAS_ACE_ARM_RESPONSE_SIZE                     1
00140
00141 #define ZRC_HA_ATTRIBUTE_STATUS_TABLE_SIZE
00142   \
00143   (ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE_ID_SIZE * 16 +
00144   ZRC_HA_SCENE_LOCAL_SCENE_VALID_ID_SIZE * 16 +
00145   ZRC_HA_ON_OFF_ON_OFF_ID_SIZE +
00146   ZRC_HA_LEVEL_CONTROL_CURRENT_LEVEL_ID_SIZE +
00147   ZRC_HA_DOOR_LOCK_LOCK_DOOR_RESPONSE_ID_SIZE +
00148   ZRC_HA_DOOR_LOCK_UNLOCK_DOOR_RESPONSE_ID_SIZE +
00149   ZRC_HA_DOOR_LOCK_TOGGLE_RESPONSE_ID_SIZE +
00150   ZRC_HA_WINDOW_COVERING_CURRENT_POSITION_LIFT_PERCENTAGE_ID_SIZE +
00151   ZRC_HA_WINDOW_COVERING_CURRENT_POSITION_TILT_PERCENTAGE_ID_SIZE +
00152   ZRC_HA_THERMOSTAT_LOCAL_TEMPERATURE_ID_SIZE +
00153   ZRC_HA_THERMOSTAT_OCCUPIED_COOLING_SETPOINT_ID_SIZE +
00154   ZRC_HA_THERMOSTAT_OCCUPIED_HEATING_SETPOINT_ID_SIZE +
00155   ZRC_HA_COLOR_CONTROL_CURRENT_HUE_ID_SIZE +
00156   ZRC_HA_COLOR_CONTOL_CURRENT_SATURATION_ID_SIZE +
00157   ZRC_HA_COLOR_CONTOL_CURRENT_X_ID_SIZE +
00158   ZRC_HA_COLOR_CONTOL_CURRENT_Y_ID_SIZE +
00159   ZRC_HA_COLOR_CONTOL_COLOR_MODE_ID_SIZE +
00160   ZRC_HA_IAS_ACE_ARM_RESPONSE_ID_SIZE)
00161
00162 #define ZRC_HA_ATTRIBUTE_TABLE_SIZE
00163   \
00164   (ZRC_HA_SCENE_STORE_LOCAL_SCENE_RESPONSE_SIZE * 16 +
00165   ZRC_HA_SCENE_LOCAL_SCENE_VALID_SIZE * 16 +
00166   ZRC_HA_LEVEL_CONTROL_CURRENT_LEVEL_SIZE +

```

```

00167 \
00168 \
00169 \
00170 \
00171 \
00172 \
00173 \
00174 \
00175 \
00176 \
00177 \
00178 \
00179 \
00180 \
00181 \
00182
00183 #define HA_ATTRIBUTE_STATUS_LENGTH           1
00184 // #define HA_ATTRIBUTE_STATUS_OFFSET        0
00185 // #define HA_ATTRIBUTE_VALUE_OFFSET          1
00186
00187 // The status byte for each HA attribute is defined by the following.
00188 #define HA_ATTRIBUTE_STATUS_CHANGED_FLAG      0x01
00189
00190
00191 typedef struct
00192 {
00193     EmberOutgoingMessageType type;
00194     uint16_t indexOrDestination;
00195     uint8_t sourceEndpoint;
00196     uint8_t destinationEndpoint;
00197 } DestStruct;
00198
00199 typedef struct
00200 {
00201     uint8_t id;
00202     uint8_t length;
00203 } HaAttributesInfo;
00204
00205 //typedef struct
00206 //{
00207 //    uint8_t stat[ZRC_HA_ATTRIBUTE_STATUS_TABLE_SIZE/8+1];
00208 //} HaAttributesStat;
00209
00210 //typedef struct
00211 //{
00212 //    uint8_t val[ZRC_HA_ATTRIBUTE_TABLE_SIZE];
00213 //} HaAttributesVal;
00214
00215
00216
00220 void emberAfPluginRf4ceZrc20HaServerClearAllHaAttributes
00221     (void);
00222
00237 EmberAfRf4ceGdpAttributeStatus
00238     emberAfPluginRf4ceZrc20HaServerGetHaAttribute
00239         (uint8_t pairingIndex,
00240         uint8_t haInstanceId,
00241         uint8_t haAttributeId,
00242         EmberAfRf4ceZrcHomeAutomationAttribute *
00243             haAttribute);
00241

```

```

00242
00257 EmberAfRf4ceGdpAttributeStatus
00258     emberAfPluginRf4ceZrc20HaServerSetHaAttribute
00259     (uint8_t pairingIndex,
00260      uint8_t haInstanceId,
00261      uint8_t haAttributeId,
00262      EmberAfRf4ceZrcHomeAutomationAttribute *
00263      haAttribute);
00264
00265 EmberStatus emberAfPluginRf4ceZrc20HaMappingToLogicalDeviceAdd
00266     (uint8_t pairingIndex,
00267      uint8_t haInstanceId,
00268      uint8_t destIndex);
00269
00270 EmberStatus emberAfPluginRf4ceZrc20HaMappingToLogicalDeviceRemove
00271     (uint8_t destIndex);
00272
00273 EmberStatus emberAfPluginRf4ceZrc20HaMappingToLogicalDeviceGet
00274     (uint8_t pairingIndex,
00275      uint8_t haInstanceId,
00276      uint8_t destIndex);
00277
00278 EmberStatus emberAfPluginRf4ceZrc20HaLogicalDeviceDestinationAdd
00279     (DestStruct* dest,
00280      uint8_t* index
00281 );
00282
00283 EmberStatus emberAfPluginRf4ceZrc20HaLogicalDeviceDestinationRemove
00284     (DestStruct* dest,
00285      uint8_t* index);
00286
00287 uint8_t emberAfPluginRf4ceZrc20HaLogicalDeviceDestinationTableSize
00288     (void);
00289
00290 EmberStatus emberAfPluginRf4ceZrc20HaLogicalDeviceDestinationGet
00291     (uint8_t pairingIndex,
00292      uint8_t haInstanceId,
00293      DestStruct*
00294      * dest);
00295
00296 EmberStatus emberAfPluginRf4ceZrc20HaLogicalDeviceIndexLookUp
00297     (DestStruct* dest,
00298      uint8_t* index);
00299
00300 void
00301     emberAfPluginRf4ceZrc20HaLogicalDeviceAndInstanceToLogicalDeviceMappingClear
00302     (void);
00303
00304 EmberStatus
00305     emberAfPluginRf4ceZrc20HaLogicalDeviceAndInstanceToLogicalDeviceMappingAdd
00306     (uint8_t pairingIndex,
00307      uint8_t haInstanceId,
00308      DestStruct* dest);
00309
00310 EmberStatus
00311     emberAfPluginRf4ceZrc20HaLogicalDeviceAndInstanceToLogicalDeviceMappingRemove
00312     (DestStruct* dest);
00313
00314

```

```

00419
00420
00421 EmberStatus emAfRf4ceZrc20ParseHaActionAndForwardToZclNetwork
    (const EmberAfRf4ceZrcActionRecord *record);
00422 void emAfRf4ceZrc20ClearLogicalDevicesTable
    (void);
00423 void emAfRf4ceZrc20ClearInstanceToLogicalDeviceTable
    (void);
00424 EmberStatus emAfRf4ceZrc20AddLogicalDeviceDestination
    (DestStruct* dest,
00425                                     uint8_t* index);
00426 EmberStatus emAfRf4ceZrc20RemoveLogicalDeviceDestination
    (uint8_t destIndex);
00427
00428 uint8_t GetLogicalDeviceDestination(uint8_t i,
00429                                         DestStruct* dest);
00430 void DestLookup(uint8_t pairingIndex,
00431                   uint8_t haInstanceId,
00432                   DestStruct* dest);
00433
00434
00435
00436 #endif // __RF4CE_ZRC20_HA_SERVER_H__
00437
00438 // END rf4ce-zrc20-ha-server

```

8.70 rf4ce-zrc20-internal.h File Reference

Macros

- #define APLC_MAX_CONFIG_WAIT_TIME_MS
- #define APLC_MAX_RESPONSE_WAIT_TIME_MS
- #define APLC_MAX_ACTION_REPEAT_TRIGGER_INTERVAL_MS
- #define APLC_SHORT_RETRY_DURATION_MS
- #define emAfRf4ceZrcClearActionCode
- #define emAfRf4ceZrcReadActionCode
- #define emAfRf4ceZrcSetActionCode
- #define emAfRf4ceZrcExchangeActionBanks(originatorCapabilities,recipientCapabilities)
- #define ACTION_CODES_SUPPORTED_RECORDS_MAX
- #define ACTION_TYPE_MASK
- #define MODIFIER_BITS_MASK
- #define MODIFIER_BITS_SPECIAL_MASK
- #define MODIFIER_BITS_SPECIAL_MARK
- #define ZRC_VERSION_NONE
- #define ZRC_VERSION_1_1
- #define ZRC_VERSION_2_0
- #define ACTION_RECORD_ACTION_CONTROL_OFFSET
- #define ACTION_RECORD_ACTION_CONTROL_LENGTH
- #define ACTION_RECORD_ACTION_CONTROL_ACTION_TYPE_MASK
- #define ACTION_RECORD_ACTION_CONTROL_ACTION_TYPE_OFFSET
- #define ACTION_RECORD_ACTION_CONTROL_MODIFIER_BITS_MASK
- #define ACTION_RECORD_ACTION_CONTROL_MODIFIER_BITS_OFFSET
- #define ACTION_RECORD_ACTION_PAYLOAD_LENGTH_OFFSET
- #define ACTION_RECORD_ACTION_PAYLOAD_LENGTH_LENGTH
- #define ACTION_RECORD_ACTION_BANK_OFFSET
- #define ACTION_RECORD_ACTION_BANK_LENGTH
- #define ACTION_RECORD_ACTION_CODE_OFFSET
- #define ACTION_RECORD_ACTION_CODE_LENGTH

- #define ACTION_RECORD_ACTION_VENDOR_OFFSET
- #define ACTION_RECORD_ACTION_VENDOR_LENGTH
- #define ZRC11_MAX_USER_CONTROL_COMMAND_PAYLOAD_LENGTH
- #define ZRC11_MAX_USER_CONTROL_COMMAND_LENGTH
- #define ZRC11_MAX_RESPONSE_WAIT_TIME
- #define ZRC_HEADER_LENGTH
- #define ZRC_HEADER_FRAME_CONTROL_OFFSET
- #define ZRC_HEADER_FRAME_CONTROL_COMMAND_CODE_MASK
- #define ZRC_PAYLOAD_OFFSET
- #define USER_CONTROL_PRESSED_LENGTH
- #define USER_CONTROL_PRESSED_RC_COMMAND_CODE_OFFSET
- #define USER_CONTROL_PRESSED_RC_COMMAND_PAYLOAD_OFFSET
- #define USER_CONTROL_REPEATED_1_0_LENGTH
- #define USER_CONTROL_REPEATED_1_1_LENGTH
- #define USER_CONTROL_REPEATED_1_1_RC_COMMAND_CODE_OFFSET
- #define USER_CONTROL_REPEATED_1_1_RC_COMMAND_PAYLOAD_OFFSET
- #define USER_CONTROL_RELEASED_1_0_LENGTH
- #define USER_CONTROL_RELEASED_1_1_LENGTH
- #define USER_CONTROL_RELEASED_1_1_RC_COMMAND_CODE_OFFSET
- #define COMMAND_DISCOVERY_REQUEST_LENGTH
- #define COMMANDS_SUPPORTED_LENGTH
- #define COMMAND_DISCOVERY_RESPONSE_LENGTH
- #define COMMAND_DISCOVERY_RESPONSE_COMMANDS_SUPPORTED_OFFSET
- #define CLIENT_NOTIFICATION_SUBTYPE_REQUEST_ACTION_MAPPING_NEGOTIATION_N
- #define CLIENT_NOTIFICATION_SUBTYPE_REQUEST_HA_PULL
- #define CLIENT_NOTIFICATION_SUBTYPE_REQUEST_SELECTIVE_ACTION_MAPPING_UPDATE
- #define CLIENT_NOTIFICATION_REQUEST_ACTION_MAPPING_NEGOTIATION_PAYLOAD_LENGTH
- #define CLIENT_NOTIFICATION_REQUEST_HA_PULL_PAYLOAD_LENGTH
- #define CLIENT_NOTIFICATION_REQUEST_HA_PULL_HA_INSTANCE_ID_OFFSET
- #define CLIENT_NOTIFICATION_REQUEST_HA_PULL_HA_INSTANCE_ID_LENGTH
- #define CLIENT_NOTIFICATION_REQUEST_HA_PULL_HA_ATTRIBUTE_DIRTY_FLAGS_OFFSET
- #define CLIENT_NOTIFICATION_REQUEST_HA_PULL_HA_ATTRIBUTE_DIRTY_FLAGS_LENGTH
- #define CLIENT_NOTIFICATION_REQUEST_SELECTIVE_AM_UPDATE_INDEX_LIST_LENGTH_OFFSET
- #define CLIENT_NOTIFICATION_REQUEST_SELECTIVE_AM_UPDATE_INDEX_LIST_LENGTH
- #define CLIENT_NOTIFICATION_REQUEST_SELECTIVE_AM_UPDATE_INDEX_LIST_OFFSET
- #define ZRC_STATE_INITIAL
- #define ZRC_STATE_ORIGINATOR_PUSH_VERSION_AND_CAPABILITIES_AND_ACTION_BANKS_VERSION
- #define ZRC_STATE_ORIGINATOR_GET_VERSION_AND_CAPABILITIES_AND_ACTION_BANKS_VERSION
- #define ZRC_STATE_ORIGINATOR_GET_ACTION_BANKS_SUPPORTED_RX
- #define ZRC_STATE_ORIGINATOR_PUSH_ACTION_BANKS_SUPPORTED_TX
- #define ZRC_STATE_ORIGINATOR_GET_ACTION_CODES_SUPPORTED_RX

- #define `ZRC_STATE_ORIGINATOR_PUSH_ACTION_CODES_SUPPORTED_TX`
- #define `ZRC_STATE_ORIGINATOR_CONFIGURATION_COMPLETE`
- #define `ZRC_STATE_RECIPIENT_PUSH_VERSION_AND_CAPABILITIES_AND_ACTION_BANKS_VERSION`
- #define `ZRC_STATE_RECIPIENT_GET_VERSION_AND_CAPABILITIES_AND_ACTION_BANKS_VERSION`
- #define `ZRC_STATE_RECIPIENT_GET_ACTION_BANKS_SUPPORTED_RX`
- #define `ZRC_STATE_RECIPIENT_PUSH_ACTION_BANKS_SUPPORTED_TX`
- #define `ZRC_STATE_RECIPIENT_GET_ACTION_CODES_SUPPORTED_RX`
- #define `ZRC_STATE_RECIPIENT_PUSH_ACTION_CODES_SUPPORTED_TX`
- #define `ZRC_STATE_RECIPIENT_CONFIGURATION_COMPLETE`
- #define `ZRC_STATE_AM_CLIENT_PUSHING_IRDB_VENDOR_SUPPORT_TO_SERVER`
- #define `ZRC_STATE_AM_CLIENT_PUSHING_MAPPABLE_ACTIONS_TO_SERVER`
- #define `ZRC_STATE_AM_CLIENT_PULLING_ACTION_MAPPINGS_FROM_SERVER`
- #define `ZRC_STATE_HA_ORIGINATOR_PUSHING_HA_SUPPORTED_TO_RECIPIENT`
- #define `ZRC_STATE_HA_ORIGINATOR_PULLING_HA_ATTRIBUTE_FROM_RECIPIENT`
- #define `ZRC_STATE_HA_ORIGINATOR_PULLING_HA_ATTRIBUTES_ON_REQUEST_FROM_RECIPIENT`
- #define `isZrcStateBindingOriginator()`
- #define `isZrcStateBindingRecipient()`
- #define `isZrcStateActionMappingClient()`
- #define `isZrcStateHaActionsOriginator()`
- #define `ZRC_INTERNAL_FLAGS_CAPABILITIES_MASK`
- #define `ZRC_INTERNAL_FLAGS_CAPABILITIES_OFFSET`
- #define `printState(command)`
- #define `printStateWithStatus(command, status)`
- #define `printGetAttribute(attributeId)`
- #define `printPushAttribute(attributeId)`
- #define `emAfRf4ceZrcSetState(newState)`
- #define `debugScriptCheck(reason)`

Functions

- void `emAfRf4ceZrc20StartConfigurationOriginator` (uint8_t pairingIndex)
- void `emAfRf4ceZrc20StartConfigurationRecipient` (uint8_t pairingIndex)
- void `emAfRf4ceZrcClearActionBank` (uint8_t *actionBanksSupported, `EmberAfRf4ceZrcActionBank` actionBank)
- bool `emAfRf4ceZrcReadActionBank` (const uint8_t *actionBanksSupported, `EmberAfRf4ceZrcActionBank` actionBank)
- void `emAfRf4ceZrcSetActionBank` (uint8_t *actionBanksSupported, `EmberAfRf4ceZrcActionBank` actionBank)
- bool `emAfRf4ceZrcHasRemainingActionBanks` (const uint8_t *actionBanksSupported)
- void `emAfRf4ceZrcGetExchangeableActionBanks` (const uint8_t *actionBanksSupportedTx, `EmberAfRf4ceZrcCapability` originatorCapabilities, const uint8_t *actionBanksSupportedRx, `EmberAfRf4ceZrcCapability` recipientCapabilities, uint8_t *actionBanksSupportedRxExchange, uint8_t *actionBanksSupportedTxExchange)
- uint8_t * `emAfRf4ceZrcGetActionCodesAttributePointer` (uint8_t attrId, uint16_t entryId, uint8_t pairingIndex)
- void `emAfRf4ceZrc20IncomingMessage` (uint8_t pairingIndex, uint16_t vendorId, const uint8_t *message, uint8_t messageLength)

- void [emAfRf4ceZrc20InitRecipient](#) (void)
- void [emAfRf4ceZrc20InitOriginator](#) (void)
- void [emAfRf4ceZrc20AttributesInit](#) (void)
- uint8_t [emAfRf4ceZrc20GetPeerZrcVersion](#) (uint8_t pairingIndex)
- void [emAfRf4ceZrcIncomingRequestActionMappingNegotiation](#) (void)
- void [emAfRf4ceZrcIncomingRequestSelectiveActionMappingUpdate](#) (const uint8_t *mappableActionsList, uint8_t mappableActionsListLength)
- void [emAfRf4ceZrcIncomingRequestHomeAutomationPull](#) (uint8_t haInstanceId, const uint8_t *haAttributeDirtyFlags)
- uint16_t [emAfRf4ceZrcGetRemoteNodeFlags](#) (uint8_t pairingIndex)
- void [emAfRf4ceZrcSetRemoteNodeFlags](#) (uint8_t pairingIndex, uint16_t flags)

Variables

- uint8_t [emAfZrcState](#)

8.70.1 Macro Definition Documentation

8.70.1.1 #define APLC_MAX_CONFIG_WAIT_TIME_MS

Definition at line 5 of file [rf4ce-zrc20-internal.h](#).

8.70.1.2 #define APLC_MAX_RESPONSE_WAIT_TIME_MS

Definition at line 9 of file [rf4ce-zrc20-internal.h](#).

8.70.1.3 #define APLC_MAX_ACTION_REPEAT_TRIGGER_INTERVAL_MS

Definition at line 13 of file [rf4ce-zrc20-internal.h](#).

8.70.1.4 #define APLC_SHORT_RETRY_DURATION_MS

Definition at line 16 of file [rf4ce-zrc20-internal.h](#).

8.70.1.5 #define emAfRf4ceZrcClearActionCode

Definition at line 27 of file [rf4ce-zrc20-internal.h](#).

8.70.1.6 #define emAfRf4ceZrcReadActionCode

Definition at line 28 of file [rf4ce-zrc20-internal.h](#).

8.70.1.7 #define emAfRf4ceZrcSetActionCode

Definition at line 29 of file [rf4ce-zrc20-internal.h](#).

8.70.1.8 `#define emAfRf4ceZrcExchangeActionBanks(originatorCapabilities, recipientCapabilities)`

Definition at line 31 of file [rf4ce-zrc20-internal.h](#).

8.70.1.9 `#define ACTION_CODES_SUPPORTED_RECORDS_MAX`

Definition at line 50 of file [rf4ce-zrc20-internal.h](#).

8.70.1.10 `#define ACTION_TYPE_MASK`

Definition at line 61 of file [rf4ce-zrc20-internal.h](#).

8.70.1.11 `#define MODIFIER_BITS_MASK`

Definition at line 62 of file [rf4ce-zrc20-internal.h](#).

8.70.1.12 `#define MODIFIER_BITS_SPECIAL_MASK`

Definition at line 67 of file [rf4ce-zrc20-internal.h](#).

8.70.1.13 `#define MODIFIER_BITS_SPECIAL_MARK`

Definition at line 68 of file [rf4ce-zrc20-internal.h](#).

8.70.1.14 `#define ZRC_VERSION_NONE`

Definition at line 70 of file [rf4ce-zrc20-internal.h](#).

8.70.1.15 `#define ZRC_VERSION_1_1`

Definition at line 71 of file [rf4ce-zrc20-internal.h](#).

8.70.1.16 `#define ZRC_VERSION_2_0`

Definition at line 72 of file [rf4ce-zrc20-internal.h](#).

8.70.1.17 `#define ACTION_RECORD_ACTION_CONTROL_OFFSET`

Definition at line 77 of file [rf4ce-zrc20-internal.h](#).

8.70.1.18 `#define ACTION_RECORD_ACTION_CONTROL_LENGTH`

Definition at line 78 of file [rf4ce-zrc20-internal.h](#).

8.70.1.19 #define ACTION_RECORD_ACTION_CONTROL_ACTION_TYPE_MASK

Definition at line 79 of file [rf4ce-zrc20-internal.h](#).

8.70.1.20 #define ACTION_RECORD_ACTION_CONTROL_ACTION_TYPE_OFFSET

Definition at line 80 of file [rf4ce-zrc20-internal.h](#).

8.70.1.21 #define ACTION_RECORD_ACTION_CONTROL_MODIFIER_BITS_MASK

Definition at line 82 of file [rf4ce-zrc20-internal.h](#).

8.70.1.22 #define ACTION_RECORD_ACTION_CONTROL_MODIFIER_BITS_OFFSET

Definition at line 83 of file [rf4ce-zrc20-internal.h](#).

8.70.1.23 #define ACTION_RECORD_ACTION_PAYLOAD_LENGTH_OFFSET

Definition at line 84 of file [rf4ce-zrc20-internal.h](#).

8.70.1.24 #define ACTION_RECORD_ACTION_PAYLOAD_LENGTH_LENGTH

Definition at line 85 of file [rf4ce-zrc20-internal.h](#).

8.70.1.25 #define ACTION_RECORD_ACTION_BANK_OFFSET

Definition at line 86 of file [rf4ce-zrc20-internal.h](#).

8.70.1.26 #define ACTION_RECORD_ACTION_BANK_LENGTH

Definition at line 87 of file [rf4ce-zrc20-internal.h](#).

8.70.1.27 #define ACTION_RECORD_ACTION_CODE_OFFSET

Definition at line 88 of file [rf4ce-zrc20-internal.h](#).

8.70.1.28 #define ACTION_RECORD_ACTION_CODE_LENGTH

Definition at line 89 of file [rf4ce-zrc20-internal.h](#).

8.70.1.29 #define ACTION_RECORD_ACTION_VENDOR_OFFSET

Definition at line 90 of file [rf4ce-zrc20-internal.h](#).

8.70.1.30 #define ACTION_RECORD_ACTION_VENDOR_LENGTH

Definition at line 91 of file [rf4ce-zrc20-internal.h](#).

8.70.1.31 #define ZRC11_MAX_USER_CONTROL_COMMAND_PAYLOAD_LENGTH

Definition at line 94 of file [rf4ce-zrc20-internal.h](#).

8.70.1.32 #define ZRC11_MAX_USER_CONTROL_COMMAND_LENGTH

Definition at line 95 of file [rf4ce-zrc20-internal.h](#).

8.70.1.33 #define ZRC11_MAX_RESPONSE_WAIT_TIME

Definition at line 96 of file [rf4ce-zrc20-internal.h](#).

8.70.1.34 #define ZRC_HEADER_LENGTH

Definition at line 101 of file [rf4ce-zrc20-internal.h](#).

8.70.1.35 #define ZRC_HEADER_FRAME_CONTROL_OFFSET

Definition at line 102 of file [rf4ce-zrc20-internal.h](#).

8.70.1.36 #define ZRC_HEADER_FRAME_CONTROL_COMMAND_CODE_MASK

Definition at line 103 of file [rf4ce-zrc20-internal.h](#).

8.70.1.37 #define ZRC_PAYLOAD_OFFSET

Definition at line 104 of file [rf4ce-zrc20-internal.h](#).

8.70.1.38 #define USER_CONTROL_PRESSED_LENGTH

Definition at line 109 of file [rf4ce-zrc20-internal.h](#).

8.70.1.39 #define USER_CONTROL_PRESSED_RC_COMMAND_CODE_OFFSET

Definition at line 110 of file [rf4ce-zrc20-internal.h](#).

8.70.1.40 #define USER_CONTROL_PRESSED_RC_COMMAND_PAYLOAD_OFFSET

Definition at line 111 of file [rf4ce-zrc20-internal.h](#).

8.70.1.41 #define USER_CONTROL_REPEAT_1_0_LENGTH

Definition at line 116 of file [rf4ce-zrc20-internal.h](#).

8.70.1.42 #define USER_CONTROL_REPEAT_1_1_LENGTH

Definition at line 117 of file [rf4ce-zrc20-internal.h](#).

8.70.1.43 #define USER_CONTROL_REPEAT_1_1_RC_COMMAND_CODE_OFFSET

Definition at line 118 of file [rf4ce-zrc20-internal.h](#).

8.70.1.44 #define USER_CONTROL_REPEAT_1_1_RC_COMMAND_PAYLOAD_OFFSET

Definition at line 119 of file [rf4ce-zrc20-internal.h](#).

8.70.1.45 #define USER_CONTROL_RELEASED_1_0_LENGTH

Definition at line 123 of file [rf4ce-zrc20-internal.h](#).

8.70.1.46 #define USER_CONTROL_RELEASED_1_1_LENGTH

Definition at line 124 of file [rf4ce-zrc20-internal.h](#).

8.70.1.47 #define USER_CONTROL_RELEASED_1_1_RC_COMMAND_CODE_OFFSET

Definition at line 125 of file [rf4ce-zrc20-internal.h](#).

8.70.1.48 #define COMMAND_DISCOVERY_REQUEST_LENGTH

Definition at line 129 of file [rf4ce-zrc20-internal.h](#).

8.70.1.49 #define COMMANDS_SUPPORTED_LENGTH

Definition at line 134 of file [rf4ce-zrc20-internal.h](#).

8.70.1.50 #define COMMAND_DISCOVERY_RESPONSE_LENGTH

Definition at line 135 of file [rf4ce-zrc20-internal.h](#).

8.70.1.51 #define COMMAND_DISCOVERY_RESPONSE_COMMANDS_SUPPORTED_OFFSET

Definition at line 136 of file [rf4ce-zrc20-internal.h](#).

8.70.1.52 #define CLIENT_NOTIFICATION_SUBTYPE_REQUEST_ACTION_MAPPING_NEGOTIATION

Definition at line 139 of file [rf4ce-zrc20-internal.h](#).

8.70.1.53 #define CLIENT_NOTIFICATION_SUBTYPE_REQUEST_HA_PULL

Definition at line 140 of file [rf4ce-zrc20-internal.h](#).

8.70.1.54 #define CLIENT_NOTIFICATION_SUBTYPE_REQUEST_SELECTIVE_ACTION_MAPPING_UPDATE

Definition at line 141 of file [rf4ce-zrc20-internal.h](#).

8.70.1.55 #define CLIENT_NOTIFICATION_REQUEST_ACTION_MAPPING_NEGOTIATION_PAYLOAD_LENGTH

Definition at line 144 of file [rf4ce-zrc20-internal.h](#).

8.70.1.56 #define CLIENT_NOTIFICATION_REQUEST_HA_PULL_PAYLOAD_LENGTH

Definition at line 149 of file [rf4ce-zrc20-internal.h](#).

8.70.1.57 #define CLIENT_NOTIFICATION_REQUEST_HA_PULL_HA_INSTANCE_ID_OFFSET

Definition at line 150 of file [rf4ce-zrc20-internal.h](#).

8.70.1.58 #define CLIENT_NOTIFICATION_REQUEST_HA_PULL_HA_INSTANCE_ID_LENGTH

Definition at line 151 of file [rf4ce-zrc20-internal.h](#).

8.70.1.59 #define CLIENT_NOTIFICATION_REQUEST_HA_PULL_HA_ATTRIBUTE_DIRTY_FLAGS_OFFSET

Definition at line 152 of file [rf4ce-zrc20-internal.h](#).

8.70.1.60 #define CLIENT_NOTIFICATION_REQUEST_HA_PULL_HA_ATTRIBUTE_DIRTY_FLAGS_LENGTH

Definition at line 153 of file [rf4ce-zrc20-internal.h](#).

8.70.1.61 #define CLIENT_NOTIFICATION_REQUEST_SELECTIVE_AM_UPDATE_INDEX_LIST_LENGTH_OFFSET

Definition at line 159 of file [rf4ce-zrc20-internal.h](#).

8.70.1.62 #define CLIENT_NOTIFICATION_REQUEST_SELECTIVE_AM_UPDATE_INDEX_LIST_LENGTH_LENGTH

Definition at line 160 of file [rf4ce-zrc20-internal.h](#).

8.70.1.63 #define CLIENT_NOTIFICATION_REQUEST_SELECTIVE_AM_UPDATE_INDEX_LIST_OFFSET

Definition at line 161 of file [rf4ce-zrc20-internal.h](#).

8.70.1.64 #define ZRC_STATE_INITIAL

Definition at line 171 of file [rf4ce-zrc20-internal.h](#).

8.70.1.65 #define ZRC_STATE_ORIGINATOR_PUSH_VERSION_AND_CAPABILITIES_AND_ACTION_BANKS_VERSION

Definition at line 172 of file [rf4ce-zrc20-internal.h](#).

8.70.1.66 #define ZRC_STATE_ORIGINATOR_GET_VERSION_AND_CAPABILITIES_AND_ACTION_BANKS_VERSION

Definition at line 173 of file [rf4ce-zrc20-internal.h](#).

8.70.1.67 #define ZRC_STATE_ORIGINATOR_GET_ACTION_BANKS_SUPPORTED_RX

Definition at line 174 of file [rf4ce-zrc20-internal.h](#).

8.70.1.68 #define ZRC_STATE_ORIGINATOR_PUSH_ACTION_BANKS_SUPPORTED_TX

Definition at line 175 of file [rf4ce-zrc20-internal.h](#).

8.70.1.69 #define ZRC_STATE_ORIGINATOR_GET_ACTION_CODES_SUPPORTED_RX

Definition at line 176 of file [rf4ce-zrc20-internal.h](#).

8.70.1.70 #define ZRC_STATE_ORIGINATOR_PUSH_ACTION_CODES_SUPPORTED_TX

Definition at line 177 of file [rf4ce-zrc20-internal.h](#).

8.70.1.71 #define ZRC_STATE_ORIGINATOR_CONFIGURATION_COMPLETE

Definition at line 178 of file [rf4ce-zrc20-internal.h](#).

8.70.1.72 #define ZRC_STATE_RECIPIENT_PUSH_VERSION_AND_CAPABILITIES_AND_ACTION_BANKS_VERSION

Definition at line 179 of file [rf4ce-zrc20-internal.h](#).

8.70.1.73 #define ZRC_STATE_RECIPIENT_GET_VERSION_AND_CAPABILITIES_AND_ACTION_BANKS_VERSION

Definition at line 180 of file [rf4ce-zrc20-internal.h](#).

8.70.1.74 #define ZRC_STATE_RECIPIENT_GET_ACTION_BANKS_SUPPORTED_RX

Definition at line 181 of file [rf4ce-zrc20-internal.h](#).

8.70.1.75 #define ZRC_STATE_RECIPIENT_PUSH_ACTION_BANKS_SUPPORTED_TX

Definition at line 182 of file [rf4ce-zrc20-internal.h](#).

8.70.1.76 #define ZRC_STATE_RECIPIENT_GET_ACTION_CODES_SUPPORTED_RX

Definition at line 183 of file [rf4ce-zrc20-internal.h](#).

8.70.1.77 #define ZRC_STATE_RECIPIENT_PUSH_ACTION_CODES_SUPPORTED_TX

Definition at line 184 of file [rf4ce-zrc20-internal.h](#).

8.70.1.78 #define ZRC_STATE_RECIPIENT_CONFIGURATION_COMPLETE

Definition at line 185 of file [rf4ce-zrc20-internal.h](#).

8.70.1.79 #define ZRC_STATE_AM_CLIENT_PUSHING_IRDB_VENDOR_SUPPORT_TO_SERVER

Definition at line 186 of file [rf4ce-zrc20-internal.h](#).

8.70.1.80 #define ZRC_STATE_AM_CLIENT_PUSHING_MAPPABLE_ACTIONS_TO_SERVER

Definition at line 187 of file [rf4ce-zrc20-internal.h](#).

8.70.1.81 #define ZRC_STATE_AM_CLIENT_PULLING_ACTION_MAPPINGS_FROM_SERVER

Definition at line 188 of file [rf4ce-zrc20-internal.h](#).

8.70.1.82 #define ZRC_STATE_HA_ORIGINATOR_PUSHING_HA_SUPPORTED_TO_RECIPIENT

Definition at line 189 of file [rf4ce-zrc20-internal.h](#).

8.70.1.83 #define ZRC_STATE_HA_ORIGINATOR_PULLING_HA_ATTRIBUTE_FROM_RECIPIENT

Definition at line 190 of file [rf4ce-zrc20-internal.h](#).

8.70.1.84 #define ZRC_STATE_HA_ORIGINATOR_PULLING_HA_ATTRIBUTES_ON_REQUEST_FROM_RECIPIENT

Definition at line 191 of file [rf4ce-zrc20-internal.h](#).

8.70.1.85 #define isZrcStateBindingOriginator()

Definition at line 195 of file [rf4ce-zrc20-internal.h](#).

8.70.1.86 #define isZrcStateBindingRecipient()

Definition at line 199 of file [rf4ce-zrc20-internal.h](#).

8.70.1.87 #define isZrcStateActionMappingClient()

Definition at line 203 of file [rf4ce-zrc20-internal.h](#).

8.70.1.88 #define isZrcStateHaActionsOriginator()

Definition at line 207 of file [rf4ce-zrc20-internal.h](#).

8.70.1.89 #define ZRC_INTERNAL_FLAGS_CAPABILITIES_MASK

Definition at line 220 of file [rf4ce-zrc20-internal.h](#).

8.70.1.90 #define ZRC_INTERNAL_FLAGS_CAPABILITIES_OFFSET

Definition at line 221 of file [rf4ce-zrc20-internal.h](#).

8.70.1.91 #define printState(*command*)

Definition at line 229 of file [rf4ce-zrc20-internal.h](#).

8.70.1.92 #define printStateWithStatus(*command*, *status*)

Definition at line 230 of file [rf4ce-zrc20-internal.h](#).

8.70.1.93 #define printGetAttribute(*attributeld*)

Definition at line 231 of file [rf4ce-zrc20-internal.h](#).

8.70.1.94 #define printPushAttribute(*attributeld*)

Definition at line 232 of file [rf4ce-zrc20-internal.h](#).

8.70.1.95 #define emAfZrcSetState(*newState*)

Definition at line 233 of file [rf4ce-zrc20-internal.h](#).

8.70.1.96 #define debugScriptCheck(*reason*)

Definition at line 242 of file [rf4ce-zrc20-internal.h](#).

8.70.2 Function Documentation

- 8.70.2.1 void emAfRf4ceZrc20StartConfigurationOriginator (uint8_t *pairingIndex*)
- 8.70.2.2 void emAfRf4ceZrc20StartConfigurationRecipient (uint8_t *pairingIndex*)
- 8.70.2.3 void emAfRf4ceZrcClearActionBank (uint8_t * *actionBanksSupported*, EmberAfRf4ceZrcActionBank *actionBank*)
- 8.70.2.4 bool emAfRf4ceZrcReadActionBank (const uint8_t * *actionBanksSupported*, EmberAfRf4ceZrcActionBank *actionBank*)
- 8.70.2.5 void emAfRf4ceZrcSetActionBank (uint8_t * *actionBanksSupported*, EmberAfRf4ceZrcActionBank *actionBank*)
- 8.70.2.6 bool emAfRf4ceZrcHasRemainingActionBanks (const uint8_t * *actionBanksSupported*)
- 8.70.2.7 void emAfRf4ceZrcGetExchangeableActionBanks (const uint8_t * *actionBanksSupportedTx*, EmberAfRf4ceZrcCapability *originatorCapabilities*, const uint8_t * *actionBanksSupportedRx*, EmberAfRf4ceZrcCapability *recipientCapabilities*, uint8_t * *actionBanksSupportedRxExchange*, uint8_t * *actionBanksSupportedTxExchange*)
- 8.70.2.8 uint8_t* emAfRf4ceZrcGetActionCodeAttributePointer (uint8_t *attrId*, uint16_t *entryId*, uint8_t *pairingIndex*)
- 8.70.2.9 void emAfRf4ceZrc20IncomingMessage (uint8_t *pairingIndex*, uint16_t *vendorId*, const uint8_t * *message*, uint8_t *messageLength*)
- 8.70.2.10 void emAfRf4ceZrc20InitRecipient (void)
- 8.70.2.11 void emAfRf4ceZrc20InitOriginator (void)
- 8.70.2.12 void emAfRf4ceZrc20AttributesInit (void)
- 8.70.2.13 uint8_t emAfRf4ceZrc20GetPeerZrcVersion (uint8_t *pairingIndex*)
- 8.70.2.14 void emAfRf4ceZrcIncomingRequestActionMappingNegotiation (void)
- 8.70.2.15 void emAfRf4ceZrcIncomingRequestSelectiveActionMappingUpdate (const uint8_t * *mappableActionsList*, uint8_t *mappableActionsListLength*)
- 8.70.2.16 void emAfRf4ceZrcIncomingRequestHomeAutomationPull (uint8_t *haInstanceId*, const uint8_t * *haAttributeDirtyFlags*)
- 8.70.2.17 uint16_t emAfRf4ceZrcGetRemoteNodeFlags (uint8_t *pairingIndex*)
- 8.70.2.18 void emAfRf4ceZrcSetRemoteNodeFlags (uint8_t *pairingIndex*, uint16_t *flags*)

8.70.3 Variable Documentation

8.70.3.1 uint8_t emAfZrcState

8.71 rf4ce-zrc20-internal.h

```

00001 // Copyright 2014 Silicon Laboratories, Inc.
00002
00003 // The maximum time the Binding Recipient shall wait to receive a command frame
00004 // from a Binding Initiator during its configuration phase.
00005 #define APLC_MAX_CONFIG_WAIT_TIME_MS 100
00006
00007 // The maximum time a node shall wait for a response command frame following a
00008 // request command frame.
00009 #define APLC_MAX_RESPONSE_WAIT_TIME_MS 100
00010
00011 // The maximum time between consecutive actions command frame transmissions
00012 // indicating a repeated action.
00013 #define APLC_MAX_ACTION_REPEAT_TRIGGER_INTERVAL_MS 200
00014
00015 // The time that an action control record should be repeated, see Table 10.
00016 #define APLC_SHORT_RETRY_DURATION_MS 100
00017
00018 void emAfRf4ceZrc20StartConfigurationOriginator
    (uint8_t pairingIndex);
00019 void emAfRf4ceZrc20StartConfigurationRecipient
    (uint8_t pairingIndex);
00020
00021 void emAfRf4ceZrcClearActionBank(uint8_t *
    actionBanksSupported,
                                         EmberAfRf4ceZrcActionBank
    actionBank);
00023 bool emAfRf4ceZrcReadActionBank(const uint8_t *
    actionBanksSupported,
                                         EmberAfRf4ceZrcActionBank
    actionBank);
00025 void emAfRf4ceZrcSetActionBank(uint8_t *
    actionBanksSupported,
                                         EmberAfRf4ceZrcActionBank
    actionBank);
00027 #define emAfRf4ceZrcClearActionCode emAfRf4ceZrcClearActionBank
00028 #define emAfRf4ceZrcReadActionCode emAfRf4ceZrcReadActionBank
00029 #define emAfRf4ceZrcSetActionCode emAfRf4ceZrcSetActionBank
00030 bool emAfRf4ceZrcHasRemainingActionBanks(
    const uint8_t *actionBanksSupported);
00031 #define emAfRf4ceZrcExchangeActionBanks(originatorCapabilities,
00032                                         recipientCapabilities) \
00033     (((originatorCapabilities) | (recipientCapabilities)) \
00034     & EMBER_AF_RF4CE_ZRC_CAPABILITY_INFORM_ABOUT_SUPPORTED_ACTIONS)
00035 void emAfRf4ceZrcGetExchangeableActionBanks
    (const uint8_t *actionBanksSupportedTx,
00036                                         EmberAfRf4ceZrcCapability
    originatorCapabilities,
00037                                         const uint8_t *
    actionBanksSupportedRx,
00038                                         EmberAfRf4ceZrcCapability
    recipientCapabilities,
00039                                         uint8_t *
    actionBanksSupportedRxExchange,
00040                                         uint8_t *
    actionBanksSupportedTxExchange);
00041 uint8_t *emAfRf4ceZrcGetActionCodesAttributePointer
    (uint8_t attrId,
00042                                         uint16_t entryId,
00043                                         uint8_t pairingIndex);
00044
00045
00046 // Each action codes supported record has a one-byte attribute id, two-byte
00047 // entry id, one-byte length, and 32-byte value. After accounting for the
00048 // overhead of attribute commands in general, there is only enough room in an
00049 // RF4CE data command for two records.
00050 #define ACTION_CODES_SUPPORTED_RECORDS_MAX 2
00051
00052 void emAfRf4ceZrc20IncomingMessage(uint8_t
    pairingIndex,
00053                                         uint16_t vendorId,

```

```

00054                                     const uint8_t *message,
00055                                     uint8_t messageLength);
00056
00057 void emAfRf4ceZrc20InitRecipient(void);
00058 void emAfRf4ceZrc20InitOriginator(void);
00059 void emAfRf4ceZrc20AttributesInit(void);
00060
00061 #define ACTION_TYPE_MASK 0x03
00062 #define MODIFIER_BITS_MASK 0xFO
00063
00064 // The action control field has the type in the lower nibble and the modifier
00065 // bits in the upper nibble. We store the type and modifiers separately, so
00066 // the lower nibble of modifiers is usable for bookkeeping purposes.
00067 #define MODIFIER_BITS_SPECIAL_MASK 0x0F
00068 #define MODIFIER_BITS_SPECIAL_MARK 0x01
00069
00070 #define ZRC_VERSION_NONE      0x00
00071 #define ZRC_VERSION_1_1       0x01
00072 #define ZRC_VERSION_2_0       0x02
00073
00074 uint8_t emAfRf4ceZrc20GetPeerZrcVersion(uint8_t
pairingIndex);
00075
00076 // Action record related macros
00077 #define ACTION_RECORD_ACTION_CONTROL_OFFSET          0
00078 #define ACTION_RECORD_ACTION_CONTROL_LENGTH           1
00079 #define ACTION_RECORD_ACTION_CONTROL_ACTION_TYPE_MASK 0x03
00080 #define ACTION_RECORD_ACTION_CONTROL_ACTION_TYPE_OFFSET 0
00081 // Bits 2-3 are reserved
00082 #define ACTION_RECORD_ACTION_CONTROL_MODIFIER_BITS_MASK 0xF0
00083 #define ACTION_RECORD_ACTION_CONTROL_MODIFIER_BITS_OFFSET 4
00084 #define ACTION_RECORD_ACTION_PAYLOAD_LENGTH_OFFSET    1
00085 #define ACTION_RECORD_ACTION_PAYLOAD_LENGTH_LENGTH    1
00086 #define ACTION_RECORD_ACTION_BANK_OFFSET              2
00087 #define ACTION_RECORD_ACTION_BANK_LENGTH             1
00088 #define ACTION_RECORD_ACTION_CODE_OFFSET             3
00089 #define ACTION_RECORD_ACTION_CODE_LENGTH            1
00090 #define ACTION_RECORD_ACTION_VENDOR_OFFSET           4
00091 #define ACTION_RECORD_ACTION_VENDOR_LENGTH          2
00092
00093 // ZRC 1.1 misc defines
00094 #define ZRC11_MAX_USER_CONTROL_COMMAND_PAYLOAD_LENGTH 4
00095 #define ZRC11_MAX_USER_CONTROL_COMMAND_LENGTH          (2 +
ZRC11_MAX_USER_CONTROL_COMMAND_PAYLOAD_LENGTH)
00096 #define ZRC11_MAX_RESPONSE_WAIT_TIME
00097
00098
00099 // ZRC header
00100 // - Frame control (1 byte)
00101 #define ZRC_HEADER_LENGTH                         1
00102 #define ZRC_HEADER_FRAME_CONTROL_OFFSET           0
00103 #define ZRC_HEADER_FRAME_CONTROL_COMMAND_CODE_MASK 0x0F
00104 #define ZRC_PAYLOAD_OFFSET                        (ZRC_HEADER_LENGTH)
00105
00106 // User Control Pressed
00107 // - RC command code (1 byte)
00108 // - RC command payload (n bytes)
00109 #define USER_CONTROL_PRESSED_LENGTH              (ZRC_HEADER_LENGTH + 1)
00110 #define USER_CONTROL_PRESSED_RC_COMMAND_CODE_OFFSET (ZRC_PAYLOAD_OFFSET)
00111 #define USER_CONTROL_PRESSED_RC_COMMAND_PAYLOAD_OFFSET (ZRC_PAYLOAD_OFFSET + 1)
00112
00113 // User Control Repeated
00114 // - RC command code (1 byte, 1.1 only)
00115 // - RC command payload (n bytes, 1.1 only)
00116 #define USER_CONTROL_REPEATED_1_0_LENGTH        (ZRC_HEADER_LENGTH)
00117 #define USER_CONTROL_REPEATED_1_1_LENGTH        (ZRC_HEADER_LENGTH + 1)
00118 #define USER_CONTROL_REPEATED_1_1_RC_COMMAND_CODE_OFFSET
(ZRC_PAYLOAD_OFFSET)
00119 #define USER_CONTROL_REPEATED_1_1_RC_COMMAND_PAYLOAD_OFFSET (ZRC_PAYLOAD_OFFSET
+ 1)
00120
00121 // User Control Released
00122 // - RC command code (1 byte, 1.1 only)
00123 #define USER_CONTROL_RELEASED_1_0_LENGTH        (ZRC_HEADER_LENGTH)
00124 #define USER_CONTROL_RELEASED_1_1_LENGTH        (ZRC_HEADER_LENGTH + 1)
00125 #define USER_CONTROL_RELEASED_1_1_RC_COMMAND_CODE_OFFSET (ZRC_PAYLOAD_OFFSET)
00126
00127 // Command Discovery Request
00128 // - Reserved (1 byte)
00129 #define COMMAND_DISCOVERY_REQUEST_LENGTH        (ZRC_HEADER_LENGTH + 1)

```

```

00130
00131 // Command Discovery Response
00132 // - Reserved (1 byte)
00133 // - Commands supported (32 bytes)
00134 #define COMMANDS_SUPPORTED_LENGTH 32
00135 #define COMMAND_DISCOVERY_RESPONSE_LENGTH (ZRC_HEADER_LENGTH + 1 +
    COMMANDS_SUPPORTED_LENGTH)
00136 #define COMMAND_DISCOVERY_RESPONSE_COMMANDS_SUPPORTED_OFFSET
    (ZRC_PAYLOAD_OFFSET + 1)
00137
00138 // Client Notification sub-types
00139 #define CLIENT_NOTIFICATION_SUBTYPE_REQUEST_ACTION_MAPPING_NEGOTIATION
    0x40
00140 #define CLIENT_NOTIFICATION_SUBTYPE_REQUEST_HA_PULL
    0x41
00141 #define CLIENT_NOTIFICATION_SUBTYPE_REQUEST_SELECTIVE_ACTION_MAPPING_UPDATE
    0x42
00142
00143 // Client Notification - Request Action Mapping Negotiation (no payload)
00144 #define CLIENT_NOTIFICATION_REQUEST_ACTION_MAPPING_NEGOTIATION_PAYLOAD_LENGTH
    0
00145
00146 // Client Notification - Request Home Automation Pull
00147 // - HA instance ID (1 byte)
00148 // - HA Attribute Dirty Flags (32 bytes)
00149 #define CLIENT_NOTIFICATION_REQUEST_HA_PULL_PAYLOAD_LENGTH
    33
00150 #define CLIENT_NOTIFICATION_REQUEST_HA_PULL_HA_INSTANCE_ID_OFFSET
    0
00151 #define CLIENT_NOTIFICATION_REQUEST_HA_PULL_HA_INSTANCE_ID_LENGTH
    1
00152 #define CLIENT_NOTIFICATION_REQUEST_HA_PULL_HA_ATTRIBUTE_DIRTY_FLAGS_OFFSET
    1
00153 #define CLIENT_NOTIFICATION_REQUEST_HA_PULL_HA_ATTRIBUTE_DIRTY_FLAGS_LENGTH
    32
00154
00155 // Client Notification - Request Selective Action Mapping Update
00156 // - Indices for Action Mapping Client to inform Action Mapping Server about
00157 // - Mappable Action Index List Length (1 byte)
00158 // - Mappable Action Index (2 bytes)*(List Length)
00159 #define
    CLIENT_NOTIFICATION_REQUEST_SELECTIVE_AM_UPDATE_INDEX_LIST_LENGTH_OFFSET 0
00160 #define
    CLIENT_NOTIFICATION_REQUEST_SELECTIVE_AM_UPDATE_INDEX_LIST_LENGTH_LENGTH 1
00161 #define CLIENT_NOTIFICATION_REQUEST_SELECTIVE_AM_UPDATE_INDEX_LIST_OFFSET
    1
00162
00163 void emAfRf4ceZrcIncomingRequestActionMappingNegotiation
    (void);
00164 void emAfRf4ceZrcIncomingRequestSelectiveActionMappingUpdate
    (const uint8_t *mappableActionsList,
00165                                     uint8_t
    mappableActionsListLength);
00166 void emAfRf4ceZrcIncomingRequestHomeAutomationPull
    (uint8_t haInstanceId,
00167                                     const uint8_t *
    haAttributeDirtyFlags);
00168
00169
00170 // Internal state machine states
00171 #define ZRC_STATE_INITIAL
    0x00
00172 #define
    ZRC_STATE_ORIGINATOR_PUSH_VERSION_AND_CAPABILITIES_AND_ACTION_BANKS_VERSION 0x01
00173 #define
    ZRC_STATE_ORIGINATOR_GET_VERSION_AND_CAPABILITIES_AND_ACTION_BANKS_VERSION 0x02
00174 #define ZRC_STATE_ORIGINATOR_GET_ACTION_BANKS_SUPPORTED_RX
    0x03
00175 #define ZRC_STATE_ORIGINATOR_PUSH_ACTION_BANKS_SUPPORTED_TX
    0x04
00176 #define ZRC_STATE_ORIGINATOR_GET_ACTION_CODES_SUPPORTED_RX
    0x05
00177 #define ZRC_STATE_ORIGINATOR_PUSH_ACTION_CODES_SUPPORTED_TX
    0x06
00178 #define ZRC_STATE_ORIGINATOR_CONFIGURATION_COMPLETE
    0x07
00179 #define
    ZRC_STATE_RECIPIENT_PUSH_VERSION_AND_CAPABILITIES_AND_ACTION_BANKS_VERSION 0x08
00180 #define
    ZRC_STATE_RECIPIENT_GET_VERSION_AND_CAPABILITIES_AND_ACTION_BANKS_VERSION 0x09

```

```

00181 #define ZRC_STATE_RECIPIENT_GET_ACTION_BANKS_SUPPORTED_RX
00182     0x0A
00183 #define ZRC_STATE_RECIPIENT_PUSH_ACTION_BANKS_SUPPORTED_TX
00184     0x0B
00185 #define ZRC_STATE_RECIPIENT_GET_ACTION_CODES_SUPPORTED_RX
00186     0x0C
00187 #define ZRC_STATE_RECIPIENT_PUSH_ACTION_CODES_SUPPORTED_TX
00188     0x0D
00189 #define ZRC_STATE_RECIPIENT_CONFIGURATION_COMPLETE
00190     0x0E
00191 #define ZRC_STATE_AM_CLIENT_PUSHING_IRDB_VENDOR_SUPPORT_TO_SERVER
00192     0x10
00193 #define ZRC_STATE_AM_CLIENT_PUSHING_MAPPABLE_ACTIONS_TO_SERVER
00194     0x20
00195 #define ZRC_STATE_AM_CLIENT_PULLING_ACTION_MAPPINGS_FROM_SERVER
00196     0x30
00197 #define ZRC_STATE_HA_ORIGINATOR_PUSHING_HA_SUPPORTED_TO_RECIPIENT
00198     0x40
00199 #define ZRC_STATE_HA_ORIGINATOR_PULLING_HA_ATTRIBUTE_FROM_RECIPIENT
00200     0x50
00201 #define ZRC_STATE_HA_ORIGINATOR_PULLING_HA_ATTRIBUTES_ON_REQUEST_FROM_RECIPIENT
00202     0x60
00203
00193 extern uint8_t emAfZrcState;
00194
00195 #define isZrcStateBindingOriginator()
00196     \
00197     (emAfZrcState >=
00198         ZRC_STATE_ORIGINATOR_PUSH_VERSION_AND_CAPABILITIES_AND_ACTION_BANKS_VERSION \
00199         && emAfZrcState <= ZRC_STATE_ORIGINATOR_CONFIGURATION_COMPLETE)
00200
00199 #define isZrcStateBindingRecipient()
00200     \
00201     (emAfZrcState >=
00202         ZRC_STATE_RECIPIENT_PUSH_VERSION_AND_CAPABILITIES_AND_ACTION_BANKS_VERSION \
00203         && emAfZrcState <= ZRC_STATE_RECIPIENT_CONFIGURATION_COMPLETE)
00204
00203 #define isZrcStateActionMappingClient()
00204     \
00205     (emAfZrcState >= ZRC_STATE_AM_CLIENT_PUSHING_IRDB_VENDOR_SUPPORT_TO_SERVER \
00206         && emAfZrcState <= ZRC_STATE_AM_CLIENT_PULLING_ACTION_MAPPINGS_FROM_SERVER)
00207
00207 #define isZrcStateHaActionsOriginator()
00208     \
00209     (emAfZrcState >= ZRC_STATE_HA_ORIGINATOR_PUSHING_HA_SUPPORTED_TO_RECIPIENT \
00210         && emAfZrcState <= ZRC_STATE_HA_ORIGINATOR_PULLING_HA_ATTRIBUTES_ON_REQUEST_FROM_RECIPIENT)
00211
00211 // We provide two implementations of these: one that stores the capabilities in
00212 // RAM (for HOST processors) and one that stores the capabilities in NVM (for
00213 // the SoC).
00214 uint16_t emAfRf4ceZrcGetRemoteNodeFlags(uint8_t
00215     pairingIndex);
00215 void emAfRf4ceZrcSetRemoteNodeFlags(uint8_t
00216     pairingIndex,
00217                                         uint16_t flags);
00218 // First byte stores the remote nodes ZRC capabilities (only the first byte of
00219 // the ZRC capabilities is used in the 2.0 version).
00220 #define ZRC_INTERNAL_FLAGS_CAPABILITIES_MASK      0x00FF
00221 #define ZRC_INTERNAL_FLAGS_CAPABILITIES_OFFSET      0
00222
00223 //###define EMBER_AF_PLUGIN_RF4CE_ZRC20_DEBUG_BINDING
00224 #if defined(EMBER_AF_PLUGIN_RF4CE_ZRC20_DEBUG_BINDING)
00225     #define emAfZrcSetState(newState) reallySetState((newState), __LINE__)
00226     #define printGetAttribute(attributeId) printAttribute(attributeId, "GET")
00227     #define printPushAttribute(attributeId) printAttribute(attributeId, "PUSH")
00228 #else
00229     #define printState(command)
00230     #define printStateWithStatus(command, status)
00231     #define printGetAttribute(attributeId)
00232     #define printPushAttribute(attributeId)
00233     #define emAfZrcSetState(newState) (emAfZrcState = (newState))
00234 #endif // EMBER_AF_PLUGIN_RF4CE_ZRC20_DEBUG_BINDING
00235
00236 #if defined(EMBER_SCRIPTED_TEST)
00237 #include "core/scripted-stub.h"
00238

```

```

00239 #define debugScriptCheck(reason)
00240     \
00240     simpleScriptCheck("scriptCheck", "scriptCheck: " reason, "")
00241 #else
00242 #define debugScriptCheck(reason)
00243 #endif // EMBER_SCRIPTED_TEST

```

8.72 rf4ce-zrc20-test.h File Reference

```

#include "../rf4ce-profile/rf4ce-profile-types.h"
#include "../rf4ce-gdp/rf4ce-gdp-types.h"
#include "rf4ce-zrc20-types.h"

```

Macros

- #define EMBER_AF_RF4CE_NODE_TYPE_TARGET
- #define EMBER_AF_PLUGIN_RF4CE_ZRC20_ACTION_REPEAT_TRIGGER_INTERVAL_MS
- #define EMBER_AF_PLUGIN_RF4CE_ZRC20_ACTION_REPEAT_WAIT_TIME_MS
- #define EMBER_AF_RF4CE_ZRC_ACTION_BANKS_RX
- #define EMBER_AF_RF4CE_ZRC_ACTION_BANKS_TX
- #define EMBER_AF_RF4CE_ZRC_ACTION_CODES_RX_COUNT
- #define EMBER_AF_RF4CE_ZRC_ACTION_CODES_RX
- #define EMBER_AF_RF4CE_ZRC_ACTION_CODES_TX_COUNT
- #define EMBER_AF_RF4CE_ZRC_ACTION_CODES_TX
- #define EMBER_AF_RF4CE_ZRC_IRDB_VENDOR_IDS
- #define EMBER_AF_RF4CE_ZRC_IRDB_VENDOR_ID_COUNT
- #define EMBER_AF_PLUGIN_RF4CE_ZRC20_REMOTE_IRDB_VENDORS_SUPPORTED_TABLE_SIZE
- #define EMBER_AF_PLUGIN_RF4CE_ZRC20_REMOTE_ACTION_CODES_TABLE_SIZE
- #define EMBER_AF_PLUGIN_RF4CE_ZRC20_ACTION_MAPPING_SERVER_MAPPABLE_ACTIONS_TABLE_SIZE
- #define EMBER_AF_PLUGIN_RF4CE_ZRC20_ACTION_MAPPING_SERVER_ACTION_MAPPINGS_HEAP_SIZE
- #define EMBER_AF_PLUGIN_RF4CE_ZRC20_ACTION_MAPPING_SERVER_ACTION_REMOTE_TABLE_SIZE
- #define EMBER_AF_RF4CE_ZRC_MAPPABLE_ACTIONS
- #define EMBER_AF_RF4CE_ZRC_MAPPABLE_ACTION_COUNT
- #define EMBER_AF_PLUGIN_RF4CE_ZRC20_ACTION_MAPPING_CLIENT_ACTION_MAPPING_HEAP_SIZE
- #define EMBER_AF_PLUGIN_RF4CE_ZRC20_ACTION_BANKS_VERSION
- #define EMBER_AF_PLUGIN_RF4CE_ZRC20_MAX_INCOMING_ACTION_RECORDS
- #define EMBER_AF_PLUGIN_RF4CE_ZRC20_MAX_OUTGOING_ACTION_RECORDS

Functions

- void emberAfPluginRf4ceZrc20LegacyCommandDiscoveryCompleteCallback (EmberStatus status, const EmberAfRf4ceZrcCommandsSupported *commandsSupported)
- void emberAfPluginRf4ceZrc20ActionCallback (const EmberAfRf4ceZrcActionRecord *record)
- void emberAfPluginRf4ceZrc20HaActionCallback (const EmberAfRf4ceZrcActionRecord *record)

- `bool emberAfPluginRf4ceProfileGdpDiscoveryRequestCallback` (const EmberEUI64 ieeeAddr, uint8_t nodeCapabilities, const EmberRf4ceVendorInfo *vendorInfo, const EmberRf4ceApplicationInfo *appInfo, uint8_t searchDevType, uint8_t rxLinkQuality)
- `bool emberAfPluginRf4ceProfileGdpDiscoveryResponseCallback` (bool atCapacity, uint8_t channel, EmberPanId panId, const EmberEUI64 ieeeAddr, uint8_t nodeCapabilities, const EmberRf4ceVendorInfo *vendorInfo, const EmberRf4ceApplicationInfo *appInfo, uint8_t rxLinkQuality, uint8_t discRequestLqi)
- `void emberAfPluginRf4ceProfileGdpDiscoveryCompleteCallback` (EmberStatus status)
- `void emberAfPluginRf4ceProfileGdpAutoDiscoveryResponseCompleteCallback` (EmberStatus status, const EmberEUI64 srcIeeeAddr, uint8_t nodeCapabilities, const EmberRf4ceVendorInfo *vendorInfo, const EmberRf4ceApplicationInfo *appInfo, uint8_t searchDevType)
- `bool emberAfPluginRf4ceProfileGdpPairRequestCallback` (EmberStatus status, uint8_t pairingIndex, const EmberEUI64 sourceIeeeAddr, uint8_t nodeCapabilities, const EmberRf4ceVendorInfo *vendorInfo, const EmberRf4ceApplicationInfo *appInfo, uint8_t keyExchangeTransferCount)
- `void emberAfPluginRf4ceProfileGdpPairCompleteCallback` (EmberStatus status, uint8_t pairingIndex, const EmberRf4ceVendorInfo *vendorInfo, const EmberRf4ceApplicationInfo *appInfo)
- `void emberAfPluginRf4ceGdpZrc20BindingCompleteCallback` (EmberAfRf4ceGdpBindingStatus status, uint8_t pairingIndex)
- `void emberAfPluginRf4ceZrc20ActionMappingsNegotiationCompleteCallback` (EmberStatus status)
- `void emberAfPluginRf4ceZrc20IncomingMappableActionCallback` (uint8_t pairingIndex, uint16_t entryIndex, EmberAfRf4ceZrcMappableAction *mappableAction)
- `uint16_t emberAfPluginRf4ceZrc20GetMappableActionCountCallback` (uint8_t pairingIndex)
- `EmberStatus emberAfPluginRf4ceZrc20GetMappableActionCallback` (uint8_t pairingIndex, uint16_t entryIndex, EmberAfRf4ceZrcMappableAction *mappableAction)
- `void emberAfPluginRf4ceZrc20IncomingActionMappingCallback` (uint8_t pairingIndex, uint16_t entryIndex, EmberAfRf4ceZrcActionMapping *actionMapping)
- `EmberStatus emberAfPluginRf4ceZrc20GetActionMappingCallback` (uint8_t pairingIndex, uint16_t entryIndex, EmberAfRf4ceZrcActionMapping *actionMapping)
- `void emberAfPluginRf4ceZrc20HomeAutomationSupportedAnnouncementCompleteCallback` (EmberStatus status)
- `void emberAfPluginRf4ceZrc20IncomingHomeAutomationSupportedCallback` (uint8_t pairingIndex, uint16_t entryIndex, EmberAfRf4ceZrcHomeAutomationSupported *haSupported)
- `EmberStatus emberAfPluginRf4ceZrc20GetHomeAutomationSupportedCallback` (uint8_t pairingIndex, uint16_t entryIndex, EmberAfRf4ceZrcHomeAutomationSupported *haSupported)
- `uint16_t emberAfPluginRf4ceZrc20GetHomeAutomationSupportedCountCallback` (uint8_t pairingIndex)
- `EmberAfRf4ceGdpAttributeStatus emberAfPluginRf4ceZrc20GetHomeAutomationAttributeCallback` (uint8_t pairingIndex, uint8_t haInstanceId, uint8_t haAttributeId, EmberAfRf4ceZrcHomeAutomationAttribute *haAttribute)
- `void emberAfPluginRf4ceZrc20PullHomeAutomationAttributeCompleteCallback` (EmberAfRf4ceGdpAttributeStatus responseStatus, EmberAfRf4ceZrcHomeAutomationAttribute *haAttribute)

8.72.1 Macro Definition Documentation

8.72.1.1 #define EMBER_AF_RF4CE_NODE_TYPE_TARGET

Definition at line 7 of file [rf4ce-zrc20-test.h](#).

8.72.1.2 #define EMBER_AF_PLUGIN_RF4CE_ZRC20_ACTION_REPEAT_TRIGGER_INTERVAL_MS

Definition at line 9 of file [rf4ce-zrc20-test.h](#).

8.72.1.3 #define EMBER_AF_PLUGIN_RF4CE_ZRC20_ACTION_REPEAT_WAIT_TIME_MS

Definition at line 10 of file [rf4ce-zrc20-test.h](#).

8.72.1.4 #define EMBER_AF_RF4CE_ZRC_ACTION_BANKS_RX

Definition at line 12 of file [rf4ce-zrc20-test.h](#).

8.72.1.5 #define EMBER_AF_RF4CE_ZRC_ACTION_BANKS_TX

Definition at line 17 of file [rf4ce-zrc20-test.h](#).

8.72.1.6 #define EMBER_AF_RF4CE_ZRC_ACTION_CODES_RX_COUNT

Definition at line 23 of file [rf4ce-zrc20-test.h](#).

8.72.1.7 #define EMBER_AF_RF4CE_ZRC_ACTION_CODES_RX

Definition at line 24 of file [rf4ce-zrc20-test.h](#).

8.72.1.8 #define EMBER_AF_RF4CE_ZRC_ACTION_CODES_TX_COUNT

Definition at line 34 of file [rf4ce-zrc20-test.h](#).

8.72.1.9 #define EMBER_AF_RF4CE_ZRC_ACTION_CODES_TX

Definition at line 35 of file [rf4ce-zrc20-test.h](#).

8.72.1.10 #define EMBER_AF_RF4CE_ZRC_IRDB_VENDOR_IDS

Definition at line 45 of file [rf4ce-zrc20-test.h](#).

8.72.1.11 #define EMBER_AF_RF4CE_ZRC_IRDB_VENDOR_ID_COUNT

Definition at line 46 of file [rf4ce-zrc20-test.h](#).

8.72.1.12 #define EMBER_AF_PLUGIN_RF4CE_ZRC20_REMOTE_IRDB_VENDORS_SUPPORTED_TABLE_SIZE

Definition at line 48 of file [rf4ce-zrc20-test.h](#).

8.72.1.13 #define EMBER_AF_PLUGIN_RF4CE_ZRC20_REMOTE_ACTION_CODES_TABLE_SIZE

Definition at line 50 of file [rf4ce-zrc20-test.h](#).

8.72.1.14 #define EMBER_AF_PLUGIN_RF4CE_ZRC20_ACTION_MAPPING_SERVER_MAPPABLE_ACTIONS_TABLE_SIZE

Definition at line [52](#) of file [rf4ce-zrc20-test.h](#).

8.72.1.15 #define EMBER_AF_PLUGIN_RF4CE_ZRC20_ACTION_MAPPING_SERVER_ACTION_MAPPINGS_H-EAP_SIZE

Definition at line [53](#) of file [rf4ce-zrc20-test.h](#).

8.72.1.16 #define EMBER_AF_PLUGIN_RF4CE_ZRC20_ACTION_MAPPING_SERVER_ACTION_REMAP_TABLE_SIZE

Definition at line [54](#) of file [rf4ce-zrc20-test.h](#).

8.72.1.17 #define EMBER_AF_RF4CE_ZRC_MAPPABLE_ACTIONS

Definition at line [56](#) of file [rf4ce-zrc20-test.h](#).

8.72.1.18 #define EMBER_AF_RF4CE_ZRC_MAPPABLE_ACTION_COUNT

Definition at line [74](#) of file [rf4ce-zrc20-test.h](#).

8.72.1.19 #define EMBER_AF_PLUGIN_RF4CE_ZRC20_ACTION_MAPPING_CLIENT_ACTION_MAPPING_HEADER_SIZE

Definition at line [76](#) of file [rf4ce-zrc20-test.h](#).

8.72.1.20 #define EMBER_AF_PLUGIN_RF4CE_ZRC20_ACTION_BANKS_VERSION

Definition at line [78](#) of file [rf4ce-zrc20-test.h](#).

8.72.1.21 #define EMBER_AF_PLUGIN_RF4CE_ZRC20_MAX_INCOMING_ACTION_RECORDS

Definition at line [80](#) of file [rf4ce-zrc20-test.h](#).

8.72.1.22 #define EMBER_AF_PLUGIN_RF4CE_ZRC20_MAX_OUTGOING_ACTION_RECORDS

Definition at line [81](#) of file [rf4ce-zrc20-test.h](#).

8.72.2 Function Documentation

8.72.2.1 void emberAfPluginRf4ceZrc20IncomingHomeAutomationSupportedCallback (uint8_t pairingIndex, uint16_t entryIndex, EmberAfRf4ceZrcHomeAutomationSupported * haSupported)

8.72.2.2 EmberStatus emberAfPluginRf4ceZrc20GetHomeAutomationSupportedCallback (uint8_t pairingIndex, uint16_t entryIndex, EmberAfRf4ceZrcHomeAutomationSupported * haSupported)

8.72.2.3 uint16_t emberAfPluginRf4ceZrc20GetHomeAutomationSupportedCountCallback (uint8_t pairingIndex)

8.73 rf4ce-zrc20-test.h

```

00001 // defines that are generated by app framework.
00002
00003 #include "../rf4ce-profile/rf4ce-profile-types.h"
00004 #include "../rf4ce-gdp/rf4ce-gdp-types.h"
00005 #include "rf4ce-zrc20-types.h"
00006
00007 #define EMBER_AF_RF4CE_NODE_TYPE_TARGET
00008
00009 #define EMBER_AF_PLUGIN_RF4CE_ZRC20_ACTION_REPEAT_TRIGGER_INTERVAL_MS      100
00010 #define EMBER_AF_PLUGIN_RF4CE_ZRC20_ACTION_REPEAT_WAIT_TIME_MS             200
00011
00012 #define EMBER_AF_RF4CE_ZRC_ACTION_BANKS_RX
00013 {0x81, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
00014   0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
00015   0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
00016   0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00}
00017 #define EMBER_AF_RF4CE_ZRC_ACTION_BANKS_TX
00018 {0x81, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
00019   0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
00020   0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
00021   0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00}
00022
00023 #define EMBER_AF_RF4CE_ZRC_ACTION_CODES_RX_COUNT      2
00024 #define EMBER_AF_RF4CE_ZRC_ACTION_CODES_RX
00025 {{true, 0, {0x0F, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
00026   0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
00027   0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
00028   0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00}},
00029 {{true, 7, {0x0F, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
00030   0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
00031   0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
00032   0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00}}}}
00033
00034 #define EMBER_AF_RF4CE_ZRC_ACTION_CODES_TX_COUNT      2
00035 #define EMBER_AF_RF4CE_ZRC_ACTION_CODES_TX
00036 {{true, 0, {0x0F, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
00037   0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
00038   0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
00039   0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00}},
00040 {{true, 7, {0x0F, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
00041   0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
00042   0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00}}}

```

```

00043             0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00}}}
00044
00045 #define EMBER_AF_RF4CE_ZRC_IRDB_VENDOR_IDS           {0x1234, 0x5678}
00046 #define EMBER_AF_RF4CE_ZRC_IRDB_VENDOR_ID_COUNT      2
00047
00048 #define EMBER_AF_PLUGIN_RF4CE_ZRC20_REMOTE_IRDB_VENDORS_SUPPORTED_TABLE_SIZE
00049     5
00050
00051 #define EMBER_AF_PLUGIN_RF4CE_ZRC20_REMOTE_ACTION_CODES_TABLE_SIZE
00052     3
00053
00054 #define EMBER_AF_PLUGIN_RF4CE_ZRC20_ACTION_MAPPING_SERVER_MAPPABLE_ACTIONS_TABLE_SIZE 5
00055
00056 #define EMBER_AF_RF4CE_ZRC_MAPPABLE_ACTIONS { \
00057     {0x00, 0x00, 0x00}, \
00058     {0x00, 0x00, 0x01}, \
00059     {0x00, 0x00, 0x02}, \
00060     {0x00, 0x00, 0x03}, \
00061     {0x00, 0x00, 0x04}, \
00062     {0x00, 0x00, 0x05}, \
00063     {0x00, 0x00, 0x06}, \
00064     {0x00, 0x00, 0x07}, \
00065     {0x00, 0x00, 0x08}, \
00066     {0x00, 0x00, 0x09}, \
00067     {0x00, 0x00, 0x0A}, \
00068     {0x00, 0x00, 0x0B}, \
00069     {0x00, 0x00, 0x0C}, \
00070     {0x00, 0x00, 0x0D}, \
00071     {0x00, 0x00, 0x0E}, \
00072     {0x00, 0x00, 0x0F}, \
00073 }
00074 #define EMBER_AF_RF4CE_ZRC_MAPPABLE_ACTION_COUNT 16
00075
00076 #define EMBER_AF_PLUGIN_RF4CE_ZRC20_ACTION_MAPPING_CLIENT_ACTION_MAPPING_HEAP_SIZE 1024
00077
00078 #define EMBER_AF_PLUGIN_RF4CE_ZRC20_ACTION_BANKS_VERSION
00079     APL_ZRC_ACTION_BANKS_VERSION_DEFAULT
00080
00081 #define EMBER_AF_PLUGIN_RF4CE_ZRC20_MAX_INCOMING_ACTION_RECORDS 10
00082 #define EMBER_AF_PLUGIN_RF4CE_ZRC20_MAX_OUTGOING_ACTION_RECORDS 5
00083
00084 void emberAfPluginRf4ceZrc20LegacyCommandDiscoveryCompleteCallback
00085     (EmberStatus status,
00086      const EmberAfRf4ceZrcCommandsSupported *
00087      commandsSupported);
00088
00089 void emberAfPluginRf4ceZrc20ActionCallback
00090     (const EmberAfRf4ceZrcActionRecord *record);
00091
00092 void emberAfPluginRf4ceZrc20HaActionCallback
00093     (const EmberAfRf4ceZrcActionRecord *record);
00094
00095 bool emberAfPluginRf4ceProfileGdpDiscoveryRequestCallback
00096     (const EmberEUI64 ieeeAddr,
00097      uint8_t
00098      nodeCapabilities,
00099      const EmberRf4ceVendorInfo * vendorInfo,
00100      const EmberRf4ceApplicationInfo * appInfo,
00101      uint8_t
00102      searchDevType,
00103      uint8_t
00104      rxLinkQuality);
00105
00106 bool emberAfPluginRf4ceProfileGdpDiscoveryResponseCallback
00107     (bool atCapacity,
00108      uint8_t channel,
00109      EmberPanId panId,
00110      const EmberEUI64
00111      ieeeAddr,
00112      uint8_t
00113      nodeCapabilities,

```

```

00102                                     const
00103     EmberRf4ceVendorInfo * vendorInfo,
00104                                     const
00105     EmberRf4ceApplicationInfo * appInfo,
00106                                     uint8_t
00107     rxLinkQuality,
00108                                     uint8_t
00109     discRequestLqi);
00110
00111 void emberAfPluginRf4ceProfileGdpDiscoveryCompleteCallback
00112     (EmberStatus status);
00113
00114 void emberAfPluginRf4ceProfileGdpAutoDiscoveryResponseCompleteCallback
00115     (EmberStatus status,
00116      EmberEUI64 srcIeeeAddr,
00117      nodeCapabilities,
00118      EmberRf4ceVendorInfo * vendorInfo,
00119      EmberRf4ceApplicationInfo * appInfo,
00120      searchDevType);
00121
00122 bool emberAfPluginRf4ceProfileGdpPairRequestCallback
00123     (EmberStatus status,
00124      EmberRf4ceVendorInfo * vendorInfo,
00125      EmberRf4ceApplicationInfo * appInfo,
00126      nodeCapabilities,
00127      EmberEUI64 sourceIeeeAddr,
00128      pairingIndex,
00129      EmberRf4ceVendorInfo * vendorInfo,
00130      EmberRf4ceApplicationInfo * appInfo,
00131      EmberRf4ceZrc20BindingCompleteCallback
00132     (EmberAfRf4ceGdpBindingStatus status,
00133      pairingIndex);
00134
00135 void emberAfPluginRf4ceProfileGdpPairCompleteCallback
00136     (EmberStatus status,
00137      EmberRf4ceVendorInfo * vendorInfo,
00138      EmberRf4ceApplicationInfo * appInfo);
00139
00140 void emberAfPluginRf4ceZrc20ActionMappingsNegotiationCompleteCallback
00141     (EmberStatus status);
00142
00143 void emberAfPluginRf4ceZrc20IncomingMappableActionCallback
00144     (uint8_t pairingIndex,
00145      uint16_t entryIndex,
00146      EmberAfRf4ceZrcMappableAction *mappableAction);
00147
00148 uint16_t emberAfPluginRf4ceZrc20GetMappableActionCountCallback
00149     (uint8_t pairingIndex);
00150
00151 EmberStatus emberAfPluginRf4ceZrc20GetMappableActionCallback
00152     (uint8_t pairingIndex,
00153      uint16_t entryIndex,
00154      EmberAfRf4ceZrcMappableAction *mappableAction);
00155
00156 void emberAfPluginRf4ceZrc20IncomingActionMappingCallback
00157     (uint8_t pairingIndex,
00158      uint16_t entryIndex,
00159      EmberAfRf4ceZrcActionMapping *actionMapping);
00160
00161 EmberStatus emberAfPluginRf4ceZrc20GetActionMappingCallback
00162     (uint8_t pairingIndex,
00163      uint16_t entryIndex,
00164      ,

```

```

00150     EmberAfRf4ceZrcActionMapping *actionMapping);
00151
00152 void emberAfPluginRf4ceZrc20HomeAutomationSupportedAnnouncementCompleteCallback
00153     (EmberStatus status);
00154
00155 void emberAfPluginRf4ceZrc20IncomingHomeAutomationSupportedCallback
00156     (uint8_t pairingIndex,
00157         uint16_t
00158         entryIndex,
00159         EmberAfRf4ceZrcHomeAutomationSupported *
00160         haSupported);
00161
00162 EmberStatus emberAfPluginRf4ceZrc20GetHomeAutomationSupportedCallback
00163     (uint8_t pairingIndex,
00164         uint16_t
00165         entryIndex,
00166         EmberAfRf4ceZrcHomeAutomationSupported *
00167         haSupported);
00168
00169 uint16_t emberAfPluginRf4ceZrc20GetHomeAutomationSupportedCountCallback
00170     (uint8_t pairingIndex);
00171
00172 EmberAfRf4ceGdpAttributeStatus
00173 emberAfPluginRf4ceZrc20GetHomeAutomationAttributeCallback
00174     (uint8_t pairingIndex,
00175         uint8_t haInstanceId,
00176         uint8_t haAttributeId,
00177         EmberAfRf4ceZrcHomeAutomationAttribute
00178         *haAttribute);
00179
00180 void emberAfPluginRf4ceZrc20PullHomeAutomationAttributeCompleteCallback
00181     (EmberAfRf4ceGdpAttributeStatus responseStatus,
00182         EmberAfRf4ceZrcHomeAutomationAttribute *
00183         haAttribute);

```

8.74 rf4ce-zrc20-tokens.h File Reference

Macros

- #define CREATOR_PLUGIN_RF4CE_ZRC20_FLAGS

8.74.1 Macro Definition Documentation

8.74.1.1 #define CREATOR_PLUGIN_RF4CE_ZRC20_FLAGS

Definition at line 6 of file [rf4ce-zrc20-tokens.h](#).

8.75 rf4ce-zrc20-tokens.h

```

00001 // Copyright 2014 Silicon Laboratories, Inc.
00002
00003 // For each pairing entry we maintain two bytes of flags. The first byte
00004 // stores the first byte of the ZRC capabilities (bytes 1-3 are reserved).
00005 // The second byte is left for future use as well.
00006 #define CREATOR_PLUGIN_RF4CE_ZRC20_FLAGS          0x8732
00007
00008 #ifdef DEFINETOKENS
00009 DEFINE_INDEXED_TOKEN(PLUGIN_RF4CE_ZRC20_FLAGS,
0010         uint16_t,

```

```

00011          EMBER_RF4CE_PAIRING_TABLE_SIZE,
00012          0x0000)
00013 #endif

```

8.76 rf4ce-zrc20-types.h File Reference

Data Structures

- struct [EmberAfRf4ceZrcActionRecord](#)
This data structure contains the ZRC action record.
- struct [EmberAfRf4ceZrcMappableAction](#)
RF4CE ZRC Mappable Action.
- struct [EmberAfRf4ceZrcActionMapping](#)
RF4CE ZRC Action Mapping.
- struct [EmberAfRf4ceZrcHomeAutomationSupported](#)
RF4CE ZRC Home Automation supported.
- struct [EmberAfRf4ceZrcHomeAutomationAttribute](#)
RF4CE ZRC Home Automation attribute.
- struct [EmberAfRf4ceZrcCommandsSupported](#)
This data structure contains the ZRC 1.x command discovery data.

Macros

- #define [EMBER_AF_RF4CE_ZRC_ATTRIBUTE_NAMES](#)
- #define [EMBER_AF_RF4CE_ZRC_ACTION_MAPPING_MAPPING_FLAGS_RF_SPECIFIED_BIT](#)
- #define [EMBER_AF_RF4CE_ZRC_ACTION_MAPPING_MAPPING_FLAGS_IR_SPECIFIED_BIT](#)
- #define [EMBER_AF_RF4CE_ZRC_ACTION_MAPPING_MAPPING_FLAGS_RF_DESCRIPTOR_FIRST_BIT](#)
- #define [EMBER_AF_RF4CE_ZRC_ACTION_MAPPING_MAPPING_FLAGS_USE_DEFAULT_BIT](#)
- #define [EMBER_AF_RF4CE_ZRC_ACTION_MAPPING_MAPPING_FLAGS_PERMANENT_BIT](#)
- #define [RF4CE_ZRC_ACTION_MAPPING_RF_CONFIG_KEEP_TRANSMITTING_UNTIL_KEY_RELEASE](#)
- #define [RF4CE_ZRC_ACTION_MAPPING_RF_CONFIG_SHORT_RF_RETRY](#)
- #define [RF4CE_ZRC_ACTION_MAPPING_RF_CONFIG_ATOMIC_ACTION](#)
- #define [RF4CE_ZRC_ACTION_MAPPING_IR_CONFIG_VENDOR_SPECIFIC](#)
- #define [EMBER_AF_RF4CE_ZRC_COMMANDS_SUPPORTED_SIZE](#)

Typedefs

- typedef uint8_t [EmberAfRf4ceZrcActionCode](#)

Enumerations

- enum EmberAfRf4ceZrcAttributeId {
 EMBER_AF_RF4CE_ZRC_ATTRIBUTE_VERSION, EMBER_AF_RF4CE_ZRC_ATTRIBUTE_CAPABILITIES, EMBER_AF_RF4CE_ZRC_ATTRIBUTE_ACTION_REPEAT_TRIGGER_INTERVAL, EMBER_AF_RF4CE_ZRC_ATTRIBUTE_ACTION_REPEAT_WAIT_TIME, EMBER_AF_RF4CE_ZRC_ATTRIBUTE_ACTION_BANKS_SUPPORTED_RX, EMBER_AF_RF4CE_ZRC_ATTRIBUTE_ACTION_BANKS_SUPPORTED_TX, EMBER_AF_RF4CE_ZRC_ATTRIBUTE_IRDB_VENDOR_SUPPORT, EMBER_AF_RF4CE_ZRC_ATTRIBUTE_ACTION_BANKS_VERSION, EMBER_AF_RF4CE_ZRC_ATTRIBUTE_ACTION_CODES_SUPPORTED_RX, EMBER_AF_RF4CE_ZRC_ATTRIBUTE_ACTION_CODES_SUPPORTED_TX, EMBER_AF_RF4CE_ZRC_ATTRIBUTE_MAPPABLE_ACTIONS, EMBER_AF_RF4CE_ZRC_ATTRIBUTE_ACTION_MAPPING, EMBER_AF_RF4CE_ZRC_ATTRIBUTE_HOME_AUTOMATION, EMBER_AF_RF4CE_ZRC_ATTRIBUTE_HOME_AUTOMATION_SUPPORTED }
- enum EmberAfRf4ceZrcCapability {
 EMBER_AF_RF4CE_ZRC_CAPABILITY_SUPPORT_ACTIONS_ORIGINATOR, EMBER_AF_RF4CE_ZRC_CAPABILITY_SUPPORT_ACTIONS_RECIPIENT, EMBER_AF_RF4CE_ZRC_CAPABILITY_SUPPORT_HA_ACTIONS_ORIGINATOR, EMBER_AF_RF4CE_ZRC_CAPABILITY_SUPPORT_HA_ACTIONS_RECIPIENT, EMBER_AF_RF4CE_ZRC_CAPABILITY_SUPPORT_ACTION_MAPPING_CLIENT, EMBER_AF_RF4CE_ZRC_CAPABILITY_SUPPORT_ACTION_MAPPING_SERVER, EMBER_AF_RF4CE_ZRC_CAPABILITY_SUPPORT_VENDOR_SPECIFIC_IRDB_FORMATS, EMBER_AF_RF4CE_ZRC_CAPABILITY_INFORM_ABOUT_SUPPORTED_ACTIONS }
- enum EmberAfRf4ceZrcCommandCode {
 EMBER_AF_RF4CE_ZRC_COMMAND_USER_CONTROL_PRESSED, EMBER_AF_RF4CE_ZRC_COMMAND_USER_CONTROL_REPEATED, EMBER_AF_RF4CE_ZRC_COMMAND_USER_CONTROL_RELEASED, EMBER_AF_RF4CE_ZRC_COMMAND_COMMAND_DISCOVERY_REQUEST, EMBER_AF_RF4CE_ZRC_COMMAND_COMMAND_DISCOVERY_RESPONSE, EMBER_AF_RF4CE_ZRC_COMMAND_USER_CONTROL_PRESSED, EMBER_AF_RF4CE_ZRC_COMMAND_USER_CONTROL_REPEATED, EMBER_AF_RF4CE_ZRC_COMMAND_USER_CONTROL_RELEASED, EMBER_AF_RF4CE_ZRC_COMMAND_COMMAND_DISCOVERY_REQUEST, EMBER_AF_RF4CE_ZRC_COMMAND_COMMAND_DISCOVERY_RESPONSE, EMBER_AF_RF4CE_ZRC_COMMAND_ACTIONS }
- enum EmberAfRf4ceZrcActionType { EMBER_AF_RF4CE_ZRC_ACTION_TYPE_STOP, EMBER_AF_RF4CE_ZRC_ACTION_TYPE_START, EMBER_AF_RF4CE_ZRC_ACTION_TYPE_REPEAT, EMBER_AF_RF4CE_ZRC_ACTION_TYPE_ATOMIC }
- enum EmberAfRf4ceZrcModifierBit {
 EMBER_AF_RF4CE_ZRC_MODIFIER_BIT_NONE, EMBER_AF_RF4CE_ZRC_MODIFIER_BIT_GUI, EMBER_AF_RF4CE_ZRC_MODIFIER_BIT_ALT, EMBER_AF_RF4CE_ZRC_MODIFIER_BIT_SHIFT, EMBER_AF_RF4CE_ZRC_MODIFIER_BIT_CTRL }

- enum EmberAfRf4ceZrcActionBank {
 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HID_KEYBOARD_PAGE_SECTION_A, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HID_TELEPHONY_PAGE_SECTION_A, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HID_CONSUMER_PAGE_SECTION_A,
 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HID_CONSUMER_PAGE_SECTION_B, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HID_CONSUMER_PAGE_SECTION_C, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HID_GAME_CONTROLS_PAGE, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_0,
 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_1, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_2, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_3, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_4,
 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_5, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_6, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_7, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_8,
 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_9, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_10, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_11, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_12,
 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_13, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_14, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_15, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_16,
 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_17, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_18, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_19, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_20,
 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_21, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_22, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_23, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_24,
 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_25, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_26, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_27, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_28,
 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_29, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_30, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_31, EMBER_AF_RF4CE_ZRC_ACTION_BANK_VENDOR_SPECIFIC_IMPLICIT_SOURCE,
 EMBER_AF_RF4CE_ZRC_ACTION_BANK_VENDOR_SPECIFIC_IMPLICIT_RECIPIENT, EMBER_AF_RF4CE_ZRC_ACTION_BANK_VENDOR_SPECIFIC_EXPLICIT }

- enum EmberAfRf4ceZrcActionBankHdmiCecActionCode {
 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_SELECT, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_UP, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_DOWN, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_LEFT,
 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_RIGHT, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_RIGHT_UP, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_RIGHT_DOWN, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_LEFT_UP,
 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_LEFT_DOWN, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_ROOT_MENU, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_SETUP_MENU, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_CONTENTS_MENU,
 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_FAVORITE_MENU, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_EXIT, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_0, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_1,
 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_2, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_3, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_4, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_5,
 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_6, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_7, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_8, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_9,
 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_DOT, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_ENTER, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_CLEAR, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_NEXT_FAVORITE,
 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_CHANNEL_UP, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_CHANNEL_DOWN, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_PREVIOUS_CHANNEL, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_SOUND_SELECT,
 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_INPUT_SELECT, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_DISPLAY_INFORMATION, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_HELP, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_PAGE_UP,
 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_PAGE_DOWN, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_POWER, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_VOLUME_UP, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_VOLUME_DOWN,
 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_MUTE, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_PLAY, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_STOP, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_PAUSE,
 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_RECORD, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_REWIND, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_FAST_FORWARD, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_EJECT,
 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_FORWARD, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_BACKWARD, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_STOP_RECORD, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_PAUSE_RECORD,
 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_ANGLE, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_SUB_PICTURE, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_VIDEO_ON_DEMAND, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_ELECTRONIC_PR
 }

- enum EmberAfRf4ceZrcActionBankHomeAutomationActionCode {
 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_SCENES_STORE_LOCAL_SCENE_0, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_SCENES_STORE_LOCAL_SCENE_1, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_SCENES_STORE_LOCAL_SCENE_2, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_SCENES_STORE_LOCAL_SCENE_3,
 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_SCENES_STORE_LOCAL_SCENE_4, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_SCENES_STORE_LOCAL_SCENE_5, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_SCENES_STORE_LOCAL_SCENE_6, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_SCENES_STORE_LOCAL_SCENE_7,
 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_SCENES_STORE_LOCAL_SCENE_8, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_SCENES_STORE_LOCAL_SCENE_9, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_SCENES_STORE_LOCAL_SCENE_10, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_SCENES_STORE_LOCAL_SCENE_11,
 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_SCENES_STORE_LOCAL_SCENE_12, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_SCENES_STORE_LOCAL_SCENE_13, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_SCENES_STORE_LOCAL_SCENE_14, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_SCENES_STORE_LOCAL_SCENE_15,
 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_SCENES_RECALL_LOCAL_SCENE_0, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_SCENES_RECALL_LOCAL_SCENE_1, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_SCENES_RECALL_LOCAL_SCENE_2, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_SCENES_RECALL_LOCAL_SCENE_3,
 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_SCENES_RECALL_LOCAL_SCENE_4, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_SCENES_RECALL_LOCAL_SCENE_5, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_SCENES_RECALL_LOCAL_SCENE_6, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_SCENES_RECALL_LOCAL_SCENE_7,
 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_SCENES_RECALL_LOCAL_SCENE_8, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_SCENES_RECALL_LOCAL_SCENE_9, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_SCENES_RECALL_LOCAL_SCENE_10, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_SCENES_RECALL_LOCAL_SCENE_11,
 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_SCENES_RECALL_LOCAL_SCENE_12, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_SCENES_RECALL_LOCAL_SCENE_13, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_SCENES_RECALL_LOCAL_SCENE_14, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_SCENES_RECALL_LOCAL_SCENE_15,
 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_ON_OFF, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_ON_OFF_ON, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_ON_OFF_TOGGLE, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_LEVEL_CONTROL_MOVE_TO_LEVEL,
 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_LEVEL_CONTROL_MOVE_TO_LEVEL,
 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_LEVEL_CONTROL_STEP, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_LEVEL_CONTROL_STOP, EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_LEVEL_CONTROL_STOP
 }

8.76.1 Macro Definition Documentation

8.76.1.1 #define EMBER_AF_RF4CE_ZRC_ATTRIBUTE_NAMES

Definition at line 32 of file [rf4ce-zrc20-types.h](#).

8.76.1.2 #define EMBER_AF_RF4CE_ZRC_ACTION_MAPPING_MAPPING_FLAGS_RF_SPECIFIED_BIT

RF4CE ZRC action mapping: Mapping Flags field.

Definition at line 378 of file [rf4ce-zrc20-types.h](#).

8.76.1.3 #define EMBER_AF_RF4CE_ZRC_ACTION_MAPPING_MAPPING_FLAGS_IR_SPECIFIED_BIT

Definition at line 379 of file [rf4ce-zrc20-types.h](#).

8.76.1.4 #define EMBER_AF_RF4CE_ZRC_ACTION_MAPPING_MAPPING_FLAGS_RF_DESCRIPTOR_FIRST_BIT

Definition at line 380 of file [rf4ce-zrc20-types.h](#).

8.76.1.5 #define EMBER_AF_RF4CE_ZRC_ACTION_MAPPING_MAPPING_FLAGS_USE_DEFAULT_BIT

Definition at line 382 of file [rf4ce-zrc20-types.h](#).

8.76.1.6 #define EMBER_AF_RF4CE_ZRC_ACTION_MAPPING_MAPPING_FLAGS_PERMANENT_BIT

Definition at line 383 of file [rf4ce-zrc20-types.h](#).

8.76.1.7 #define RF4CE_ZRC_ACTION_MAPPING_RF_CONFIG_KEEP_TRANSMITTING_UNTIL_KEY_RELEASE

RF4CE ZRC action mapping: RF Config field.

Definition at line 389 of file [rf4ce-zrc20-types.h](#).

8.76.1.8 #define RF4CE_ZRC_ACTION_MAPPING_RF_CONFIG_SHORT_RF_RETRY

Definition at line 390 of file [rf4ce-zrc20-types.h](#).

8.76.1.9 #define RF4CE_ZRC_ACTION_MAPPING_RF_CONFIG_ATOMIC_ACTION

Definition at line 391 of file [rf4ce-zrc20-types.h](#).

8.76.1.10 #define RF4CE_ZRC_ACTION_MAPPING_IR_CONFIG_VENDOR_SPECIFIC

RF4CE ZRC action mapping: IR Config field.

Definition at line 397 of file [rf4ce-zrc20-types.h](#).

8.76.1.11 #define EMBER_AF_RF4CE_ZRC_COMMANDS_SUPPORTED_SIZE

Size of the ZRC 1.x command discovery data in bytes (32).

Definition at line 437 of file [rf4ce-zrc20-types.h](#).

8.76.2 Typedef Documentation

8.76.2.1 typedef uint8_t EmberAfRf4ceZrcActionCode

Definition at line 173 of file [rf4ce-zrc20-types.h](#).

8.76.3 Enumeration Type Documentation

8.76.3.1 enum EmberAfRf4ceZrcAttributeId

RF4CE ZRC attribute ids.

Enumerator:

```
EMBER_AF_RF4CE_ZRC_ATTRIBUTE_VERSION
EMBER_AF_RF4CE_ZRC_ATTRIBUTE_CAPABILITIES
EMBER_AF_RF4CE_ZRC_ATTRIBUTE_ACTION_REPEAT_TRIGGER_INTERVAL
EMBER_AF_RF4CE_ZRC_ATTRIBUTE_ACTION_REPEAT_WAIT_TIME
EMBER_AF_RF4CE_ZRC_ATTRIBUTE_ACTION_BANKS_SUPPORTED_RX
EMBER_AF_RF4CE_ZRC_ATTRIBUTE_ACTION_BANKS_SUPPORTED_TX
EMBER_AF_RF4CE_ZRC_ATTRIBUTE_IRDB_VENDOR_SUPPORT
EMBER_AF_RF4CE_ZRC_ATTRIBUTE_ACTION_BANKS_VERSION
EMBER_AF_RF4CE_ZRC_ATTRIBUTE_ACTION_CODES_SUPPORTED_RX
EMBER_AF_RF4CE_ZRC_ATTRIBUTE_ACTION_CODES_SUPPORTED_TX
EMBER_AF_RF4CE_ZRC_ATTRIBUTE_MAPPABLE_ACTIONS
EMBER_AF_RF4CE_ZRC_ATTRIBUTE_ACTION_MAPPINGS
EMBER_AF_RF4CE_ZRC_ATTRIBUTE_HOME_AUTOMATION
EMBER_AF_RF4CE_ZRC_ATTRIBUTE_HOME_AUTOMATION_SUPPORTED
```

Definition at line 10 of file [rf4ce-zrc20-types.h](#).

8.76.3.2 enum EmberAfRf4ceZrcCapability

RF4CE ZRC capabilities.

Enumerator:

```
EMBER_AF_RF4CE_ZRC_CAPABILITY_SUPPORT_ACTIONS_ORIGINATOR
EMBER_AF_RF4CE_ZRC_CAPABILITY_SUPPORT_ACTIONS_RECIPIENT
EMBER_AF_RF4CE_ZRC_CAPABILITY_SUPPORT_HA_ACTIONS_ORIGINATOR
EMBER_AF_RF4CE_ZRC_CAPABILITY_SUPPORT_HA_ACTIONS_RECIPIENT
```

*EMBER_AF_RF4CE_ZRC_CAPABILITY_SUPPORT_ACTION_MAPPING_CLIENT
 EMBER_AF_RF4CE_ZRC_CAPABILITY_SUPPORT_ACTION_MAPPING_SERVER
 EMBER_AF_RF4CE_ZRC_CAPABILITY_SUPPORT_VENDOR_SPECIFIC_IRDB_FORMATS*

EMBER_AF_RF4CE_ZRC_CAPABILITY_INFORM_ABOUT_SUPPORTED_ACTIONS

Definition at line 52 of file [rf4ce-zrc20-types.h](#).

8.76.3.3 enum EmberAfRf4ceZrcCommandCode

RF4CE ZRC 2.0 command codes.

Enumerator:

*EMBER_AF_RF4CE_ZRC_COMMAND_USER_CONTROL_PRESSED
 EMBER_AF_RF4CE_ZRC_COMMAND_USER_CONTROL_REPEATED
 EMBER_AF_RF4CE_ZRC_COMMAND_USER_CONTROL_RELEASED
 EMBER_AF_RF4CE_ZRC_COMMAND_COMMAND_DISCOVERY_REQUEST
 EMBER_AF_RF4CE_ZRC_COMMAND_COMMAND_DISCOVERY_RESPONSE
 EMBER_AF_RF4CE_ZRC_COMMAND_USER_CONTROL_PRESSED
 EMBER_AF_RF4CE_ZRC_COMMAND_USER_CONTROL_REPEATED
 EMBER_AF_RF4CE_ZRC_COMMAND_USER_CONTROL_RELEASED
 EMBER_AF_RF4CE_ZRC_COMMAND_COMMAND_DISCOVERY_REQUEST
 EMBER_AF_RF4CE_ZRC_COMMAND_COMMAND_DISCOVERY_RESPONSE
 EMBER_AF_RF4CE_ZRC_COMMAND_ACTIONS*

Definition at line 72 of file [rf4ce-zrc20-types.h](#).

8.76.3.4 enum EmberAfRf4ceZrcActionType

RF4CE ZRC action banks.

Enumerator:

*EMBER_AF_RF4CE_ZRC_ACTION_TYPE_STOP
 EMBER_AF_RF4CE_ZRC_ACTION_TYPE_START
 EMBER_AF_RF4CE_ZRC_ACTION_TYPE_REPEAT
 EMBER_AF_RF4CE_ZRC_ACTION_TYPE_ATOMIC*

Definition at line 90 of file [rf4ce-zrc20-types.h](#).

8.76.3.5 enum EmberAfRf4ceZrcModifierBit

RF4CE ZRC modifier bits.

Enumerator:

EMBER_AF_RF4CE_ZRC_MODIFIER_BIT_NONE

```
EMBER_AF_RF4CE_ZRC_MODIFIER_BIT_GUI
EMBER_AF_RF4CE_ZRC_MODIFIER_BIT_ALT
EMBER_AF_RF4CE_ZRC_MODIFIER_BIT_SHIFT
EMBER_AF_RF4CE_ZRC_MODIFIER_BIT_CTRL
```

Definition at line 106 of file [rf4ce-zrc20-types.h](#).

8.76.3.6 enum EmberAfRf4ceZrcActionBank

RF4CE ZRC action banks.

Enumerator:

```
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HID_KEYBOARD_PAGE_SECTION_A
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HID_TELEPHONY_PAGE_SECTION_A
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HID_CONSUMER_PAGE_SECTION_A
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HID_CONSUMER_PAGE_SECTION_B
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HID_CONSUMER_PAGE_SECTION_C
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HID_GAME_CONTROLS_PAGE
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_0
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_1
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_2
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_3
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_4
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_5
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_6
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_7
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_8
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_9
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_10
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_11
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_12
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_13
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_14
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_15
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_16
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_17
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_18
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_19
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_20
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_21
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_22
```

EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_23
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_24
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_25
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_26
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_27
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_28
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_29
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_30
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_31
EMBER_AF_RF4CE_ZRC_ACTION_BANK_VENDOR_SPECIFIC_IMPLICIT_SOURCE
EMBER_AF_RF4CE_ZRC_ACTION_BANK_VENDOR_SPECIFIC_IMPLICIT_RECIPIENT
EMBER_AF_RF4CE_ZRC_ACTION_BANK_VENDOR_SPECIFIC_EXPLICIT

Definition at line 123 of file [rf4ce-zrc20-types.h](#).

8.76.3.7 enum EmberAfRf4ceZrcActionBankHdmiCecActionCode

RF4CE ZRC HDMI-CEC action codes.

Enumerator:

EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_SELECT
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_UP
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_DOWN
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_LEFT
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_RIGHT
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_RIGHT_UP
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_RIGHT_DOWN
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_LEFT_UP
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_LEFT_DOWN
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_ROOT_MENU
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_SETUP_MENU
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_CONTENTS_MENU

EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_FAVORITE_MENU

EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_EXIT
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_0
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_1
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_2
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_3
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_4
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_5
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_6

EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_7
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_8
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_9
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_DOT
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_ENTER
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_CLEAR
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_NEXT_FAVORITE

EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_CHANNEL_UP
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_CHANNEL_DOWN

EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_PREVIOUS_CHANNEL

EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_SOUND_SELECT

EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_INPUT_SELECT

EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_DISPLAY_INFORMATION

EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_HELP
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_PAGE_UP
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_PAGE_DOWN
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_POWER
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_VOLUME_UP
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_VOLUME_DOWN

EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_MUTE
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_PLAY
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_STOP
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_PAUSE
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_RECORD
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_REWIND
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_FAST_FORWARD

EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_EJECT
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_FORWARD
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_BACKWARD
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_STOP_RECORD

EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_PAUSE_RECORD

EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_ANGLE
EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_SUB_PICTURE

EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_VIDEO_ON_DEMAND

EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_ELECTRONIC_PROGRAM_GUIDE

EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_TIMER_PROGRAMMING

EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_INITIAL_CONFIGURATION

EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_PLAY_FUNCTION

EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_PAUSE_PLAY_FUNCTION

EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_RECORD_FUNCTION

EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_PAUSE_RECORD_FUNCTION

EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_STOP_FUNCTION

EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_MUTE_FUNCTION

EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_RESTORE_VOLUME_FUNCTION

EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_TUNE_FUNCTION

EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_SELECT_MEDIA_FUNCTION

EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_SELECT_A_V_INPUT_FUNCTION

EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_SELECT_AUDIO_INPUT_FUNCTION

EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_POWER_TOGGLE_FUNCTION

EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_POWER_OFF_FUNCTION

EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_POWER_ON_FUNCTION

EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_F1_BLUE

EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_F2_RED

EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_F3_GREEN

EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_F4_YELLOW

EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_F5

EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_DATA

Definition at line 179 of file [rf4ce-zrc20-types.h](#).

8.76.3.8 enum EmberAfRf4ceZrcActionBankHomeAutomationActionCode

RF4CE ZRC Home Automation action codes.

Enumerator:

EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_DOOR_LOCK_UNLOCK_CODE

EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_DOOR_LOCK_TOGGLE_CODE

EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_DOOR_LOCK_UNLOCK_WITH_CODE

EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_WINDOW_COVERING_UP_CODE

EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_WINDOW_COVERING_DOWN_CODE

EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_WINDOW_COVERING_STOP_CODE

EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_WINDOW_COVERING_GO_UP_CODE

EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_WINDOW_COVERING_GO_DOWN_CODE

EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_WINDOW_COVERING_GO_STOP_CODE

EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_THERMOSTAT_SETPOINT_UP_CODE

EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_COLOR_CONTROL_MOVE_UP_CODE

EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_COLOR_CONTROL_MOVE_DOWN_CODE

EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_COLOR_CONTROL_STEP_UP_CODE

EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_COLOR_CONTROL_MOVE_UP_CODE

EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_COLOR_CONTROL_STEP_DOWN_CODE

EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_COLOR_CONTROL_MOVE_DOWN_CODE

EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_COLOR_CONTROL_STEP_UP_CODE

EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_COLOR_CONTROL_MOVE_UP_CODE

EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_COLOR_CONTROL_STEP_DOWN_CODE

EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_COLOR_CONTROL_MOVE_DOWN_CODE

EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_IAS_ACE_ARM_CODE

EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_IAS_ACE_BYPASS

EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_IAS_ACE_EMERGENCY

EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_IAS_ACE_FIRE

EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_IAS_ACE_PANIC

EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_PREVIOUS_DESTINATION

EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_NEXT_DESTINATION_GROUP

Definition at line 269 of file rf4ce-zrc20-types.h.

8.77 rf4ce-zrc20-types.h

```

00001 // Copyright 2014 Silicon Laboratories, Inc.
00002
00003 #ifndef __RF4CE_ZRC20_TYPES_H__
00004 #define __RF4CE_ZRC20_TYPES_H__
00005
00009 #ifdef DOXYGEN_SHOULD_SKIP_THIS
00010 enum EmberAfRf4ceZrcAttributeId
00011 {
00012     typedef uint8_t EmberAfRf4ceZrcAttributeId;
00013     enum
00014 };
00015 {
00016     EMBER_AF_RF4CE_ZRC_ATTRIBUTE_VERSION
00017         = 0xA0,
00018     EMBER_AF_RF4CE_ZRC_ATTRIBUTE_CAPABILITIES
00019         = 0xA1,
00020     EMBER_AF_RF4CE_ZRC_ATTRIBUTE_ACTION_REPEAT_TRIGGER_INTERVAL
00021         = 0xA2,
00022     EMBER_AF_RF4CE_ZRC_ATTRIBUTE_ACTION_REPEAT_WAIT_TIME
00023         = 0xA3,
00024     EMBER_AF_RF4CE_ZRC_ATTRIBUTE_ACTION_BANKS_SUPPORTED_RX
00025         = 0xA4,
00026     EMBER_AF_RF4CE_ZRC_ATTRIBUTE_ACTION_BANKS_SUPPORTED_TX
00027         = 0xA5,
00028     EMBER_AF_RF4CE_ZRC_ATTRIBUTE_IRDB_VENDOR_SUPPORT
00029         = 0xA6,
00030     EMBER_AF_RF4CE_ZRC_ATTRIBUTE_ACTION_BANKS_VERSION
00031         = 0xA7,
00032     EMBER_AF_RF4CE_ZRC_ATTRIBUTE_ACTION_CODES_SUPPORTED_RX
00033         = 0xC0,
00034     EMBER_AF_RF4CE_ZRC_ATTRIBUTE_ACTION_CODES_SUPPORTED_TX
00035         = 0xC1,
00036     EMBER_AF_RF4CE_ZRC_ATTRIBUTE_MAPPABLE_ACTIONS
00037         = 0xC2,
00038     EMBER_AF_RF4CE_ZRC_ATTRIBUTE_ACTION_MAPPINGS
00039         = 0xC3,
00040     EMBER_AF_RF4CE_ZRC_ATTRIBUTE_HOME_AUTOMATION
00041         = 0xC4,
00042     EMBER_AF_RF4CE_ZRC_ATTRIBUTE_HOME_AUTOMATION_SUPPORTED
00043         = 0xC5,
00044 };
00045
00046 #define EMBER_AF_RF4CE_ZRC_ATTRIBUTE_NAMES
00047     \
00048     {EMBER_AF_RF4CE_ZRC_ATTRIBUTE_VERSION, "version"}, \
00049     {EMBER_AF_RF4CE_ZRC_ATTRIBUTE_CAPABILITIES, "capabilities"}, \
00050     {EMBER_AF_RF4CE_ZRC_ATTRIBUTE_ACTION_REPEAT_TRIGGER_INTERVAL, "action repeat trigger interval"}, \

```

```

00036     {EMBER_AF_RF4CE_ZRC_ATTRIBUTE_ACTION_REPEAT_WAIT_TIME,
00037         \
00038     {EMBER_AF_RF4CE_ZRC_ATTRIBUTE_ACTION_BANKS_SUPPORTED_RX,
00039         \
00040     {EMBER_AF_RF4CE_ZRC_ATTRIBUTE_ACTION_BANKS_SUPPORTED_TX,
00041         \
00042     {EMBER_AF_RF4CE_ZRC_ATTRIBUTE_IRDB_VENDOR_SUPPORT,
00043         \
00044     {EMBER_AF_RF4CE_ZRC_ATTRIBUTE_ACTION_BANKS_VERSION,
00045         \
00046     {EMBER_AF_RF4CE_ZRC_ATTRIBUTE_ACTION_CODES_SUPPORTED_RX,
00047         \
00048     {EMBER_AF_RF4CE_ZRC_ATTRIBUTE_ACTION_CODES_SUPPORTED_TX,
00049         \
00050     {EMBER_AF_RF4CE_ZRC_ATTRIBUTE_MAPPABLE_ACTIONS,
00051         \
00052     {EMBER_AF_RF4CE_ZRC_ATTRIBUTE_ACTION_MAPPINGS,
00053         \
00054     {EMBER_AF_RF4CE_ZRC_ATTRIBUTE_HOME_AUTOMATION,
00055         \
00056     {EMBER_AF_RF4CE_ZRC_ATTRIBUTE_HOME_AUTOMATION_SUPPORTED,
00057         \
00058     #ifdef DOXYGEN_SHOULD_SKIP_THIS
00059     enum EmberAfRf4ceZrcCapability
00060     #else
00061     typedef uint32_t EmberAfRf4ceZrcCapability;
00062     enum
00063     #endif
00064     {
00065         EMBER_AF_RF4CE_ZRC_CAPABILITY_SUPPORT_ACTIONS_ORIGINATOR
00066             = 0x00000001,
00067         EMBER_AF_RF4CE_ZRC_CAPABILITY_SUPPORT_ACTIONS_RECIPIENT
00068             = 0x00000002,
00069         EMBER_AF_RF4CE_ZRC_CAPABILITY_SUPPORT_HA_ACTIONS_ORIGINATOR
00070             = 0x00000004,
00071         EMBER_AF_RF4CE_ZRC_CAPABILITY_SUPPORT_HA_ACTIONS_RECIPIENT
00072             = 0x00000008,
00073         EMBER_AF_RF4CE_ZRC_CAPABILITY_SUPPORT_ACTION_MAPPING_CLIENT
00074             = 0x00000010,
00075         EMBER_AF_RF4CE_ZRC_CAPABILITY_SUPPORT_ACTION_MAPPING_SERVER
00076             = 0x00000020,
00077         EMBER_AF_RF4CE_ZRC_CAPABILITY_SUPPORT_VENDOR_SPECIFIC_IRDB_FORMATS
00078             = 0x00000040,
00079         EMBER_AF_RF4CE_ZRC_CAPABILITY_INFORM_ABOUT_SUPPORTED_ACTIONS
00080             = 0x00000080,
00081     };
00082     #ifdef DOXYGEN_SHOULD_SKIP_THIS
00083     enum EmberAfRf4ceZrcCommandCode
00084     #else
00085     typedef uint8_t EmberAfRf4ceZrcCommandCode;
00086     enum
00087     #endif
00088     {
00089         EMBER_AF_RF4CE_ZRC_COMMAND_USER_CONTROL_PRESSED
00090             = 0x01,
00091         EMBER_AF_RF4CE_ZRC_COMMAND_USER_CONTROL_REPEATED
00092             = 0x02,
00093         EMBER_AF_RF4CE_ZRC_COMMAND_USER_CONTROL_RELEASED
00094             = 0x03,
00095         EMBER_AF_RF4CE_ZRC_COMMAND_COMMAND_DISCOVERY_REQUEST
00096             = 0x04,
00097         EMBER_AF_RF4CE_ZRC_COMMAND_COMMAND_DISCOVERY_RESPONSE
00098             = 0x05,
00099         EMBER_AF_RF4CE_ZRC_COMMAND_ACTIONS
00100             = 0x06,
00101     };
00102     #ifdef DOXYGEN_SHOULD_SKIP_THIS
00103     enum EmberAfRf4ceZrcActionType
00104     #else
00105     typedef uint8_t EmberAfRf4ceZrcActionType;
00106     enum
00107     #endif
00108     {
00109         EMBER_AF_RF4CE_ZRC_ACTION_TYPE_STOP      =
00110             0x00,
00111         EMBER_AF_RF4CE_ZRC_ACTION_TYPE_START    =
00112             0x01
00113     };

```

```

0x01,
00098 EMBER_AF_RF4CE_ZRC_ACTION_TYPE_REPEAT =
0x02,
00099 EMBER_AF_RF4CE_ZRC_ACTION_TYPE_ATOMIC =
0x03,
00100 };
00101
00105 #ifdef DOXYGEN_SHOULD_SKIP_THIS
00106 enum EmberAfRf4ceZrcModifierBit
00107 #else
00108 typedef uint8_t EmberAfRf4ceZrcModifierBit;
00109 enum
00110 #endif
00111 {
00112     EMBER_AF_RF4CE_ZRC_MODIFIER_BIT_NONE =
0x00,
00113     EMBER_AF_RF4CE_ZRC_MODIFIER_BIT_GUI =
0x10,
00114     EMBER_AF_RF4CE_ZRC_MODIFIER_BIT_ALT =
0x20,
00115     EMBER_AF_RF4CE_ZRC_MODIFIER_BIT_SHIFT =
0x40,
00116     EMBER_AF_RF4CE_ZRC_MODIFIER_BIT_CTRL =
0x80,
00117 };
00118
00122 #ifdef DOXYGEN_SHOULD_SKIP_THIS
00123 enum EmberAfRf4ceZrcActionBank
00124 #else
00125 typedef uint8_t EmberAfRf4ceZrcActionBank;
00126 enum
00127 #endif
00128 {
00129     EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC
                = 0x00,
00130     EMBER_AF_RF4CE_ZRC_ACTION_BANK_HID_KEYBOARD_PAGE_SECTION_A
                = 0x20,
00131     EMBER_AF_RF4CE_ZRC_ACTION_BANK_HID_TELEPHONY_PAGE_SECTION_A
                = 0x21,
00132     EMBER_AF_RF4CE_ZRC_ACTION_BANK_HID_CONSUMER_PAGE_SECTION_A
                = 0x22,
00133     EMBER_AF_RF4CE_ZRC_ACTION_BANK_HID_CONSUMER_PAGE_SECTION_B
                = 0x23,
00134     EMBER_AF_RF4CE_ZRC_ACTION_BANK_HID_CONSUMER_PAGE_SECTION_C
                = 0x24,
00135     EMBER_AF_RF4CE_ZRC_ACTION_BANK_HID_GAME_CONTROLS_PAGE
                = 0x25,
00136     EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_0
                = 0x80,
00137     EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_1
                = 0x81,
00138     EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_2
                = 0x82,
00139     EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_3
                = 0x83,
00140     EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_4
                = 0x84,
00141     EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_5
                = 0x85,
00142     EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_6
                = 0x86,
00143     EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_7
                = 0x87,
00144     EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_8
                = 0x88,
00145     EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_9
                = 0x89,
00146     EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_10
                = 0x8A,
00147     EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_11
                = 0x8B,
00148     EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_12
                = 0x8C,
00149     EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_13
                = 0x8D,
00150     EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_14
                = 0x8E,
00151     EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_15
                = 0x8F,
00152     EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_16

```

```

        = 0x90,
00153 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_17
        = 0x91,
00154 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_18
        = 0x92,
00155 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_19
        = 0x93,
00156 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_20
        = 0x94,
00157 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_21
        = 0x95,
00158 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_22
        = 0x96,
00159 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_23
        = 0x97,
00160 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_24
        = 0x98,
00161 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_25
        = 0x99,
00162 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_26
        = 0x9A,
00163 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_27
        = 0x9B,
00164 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_28
        = 0x9C,
00165 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_29
        = 0x9D,
00166 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_30
        = 0x9E,
00167 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_INSTANCE_31
        = 0x9F,
00168 EMBER_AF_RF4CE_ZRC_ACTION_BANK_VENDOR_SPECIFIC_IMPLICIT_SOURCE
        = 0xA0,
00169 EMBER_AF_RF4CE_ZRC_ACTION_BANK_VENDOR_SPECIFIC_IMPLICIT_RECIPIENT
        = 0xC0,
00170 EMBER_AF_RF4CE_ZRC_ACTION_BANK_VENDOR_SPECIFIC_EXPLICIT
        = 0xE0,
00171 };
00172
00173 typedef uint8_t EmberAfRf4ceZrcActionCode;
00174
00175 #ifdef DOXYGEN_SHOULD_SKIP_THIS
00176 enum EmberAfRf4ceZrcActionBankHdmiCecActionCode
00177 #else
00178 typedef uint8_t EmberAfRf4ceZrcActionBankHdmiCecActionCode
00179 ;
00180 enum
00181 #endif
00182 {
00183     EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_SELECT
        = 0x00,
00184     EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_UP
        = 0x01,
00185     EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_DOWN
        = 0x02,
00186     EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_LEFT
        = 0x03,
00187     EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_RIGHT
        = 0x04,
00188     EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_RIGHT_UP
        = 0x05,
00189     EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_RIGHT_DOWN
        = 0x06,
00190     EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_LEFT_UP
        = 0x07,
00191     EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_LEFT_DOWN
        = 0x08,
00192     EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_ROOT_MENU
        = 0x09,
00193     EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_SETUP_MENU
        = 0x0A,
00194     EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_CONTENTS_MENU
        = 0x0B,
00195     EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_FAVORITE_MENU
        = 0x0C,
00196     EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_EXIT
        = 0x0D,
00197     EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_0
        = 0x20,
00198     EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_1

```

```

        = 0x21,
00201 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_2
        = 0x22,
00202 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_3
        = 0x23,
00203 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_4
        = 0x24,
00204 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_5
        = 0x25,
00205 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_6
        = 0x26,
00206 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_7
        = 0x27,
00207 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_8
        = 0x28,
00208 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_9
        = 0x29,
00209 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_DOT
        = 0x2A,
00210 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_ENTER
        = 0x2B,
00211 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_CLEAR
        = 0x2C,
00212 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_NEXT_FAVORITE
        = 0x2F,
00213 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_CHANNEL_UP
        = 0x30,
00214 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_CHANNEL_DOWN
        = 0x31,
00215 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_PREVIOUS_CHANNEL
        = 0x32,
00216 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_SOUND_SELECT
        = 0x33,
00217 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_INPUT_SELECT
        = 0x34,
00218 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_DISPLAY_INFORMATION
        = 0x35,
00219 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_HELP
        = 0x36,
00220 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_PAGE_UP
        = 0x37,
00221 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_PAGE_DOWN
        = 0x38,
00222 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_POWER
        = 0x40,
00223 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_VOLUME_UP
        = 0x41,
00224 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_VOLUME_DOWN
        = 0x42,
00225 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_MUTE
        = 0x43,
00226 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_PLAY
        = 0x44,
00227 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_STOP
        = 0x45,
00228 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_PAUSE
        = 0x46,
00229 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_RECORD
        = 0x47,
00230 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_REWIND
        = 0x48,
00231 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_FAST_FORWARD
        = 0x49,
00232 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_EJECT
        = 0x4A,
00233 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_FORWARD
        = 0x4B,
00234 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_BACKWARD
        = 0x4C,
00235 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_STOP_RECORD
        = 0x4D,
00236 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_PAUSE_RECORD
        = 0x4E,
00237 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_ANGLE
        = 0x50,
00238 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_SUB_PICTURE
        = 0x51,
00239 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_VIDEO_ON_DEMAND
        = 0x52,
00240 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_ELECTRONIC_PROGRAM_GUIDE

```

```

        = 0x53,
00241 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_TIMER_PROGRAMMING
        = 0x54,
00242 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_INITIAL_CONFIGURATION
        = 0x55,
00243 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_PLAY_FUNCTION
        = 0x60, // Play Mode - 1 byte
00244 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_PAUSE_PLAY_FUNCTION
        = 0x61,
00245 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_RECORD_FUNCTION
        = 0x62,
00246 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_PAUSE_RECORD_FUNCTION
        = 0x63,
00247 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_STOP_FUNCTION
        = 0x64,
00248 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_MUTE_FUNCTION
        = 0x65,
00249 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_RESTORE_VOLUME_FUNCTION
        = 0x66,
00250 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_TUNE_FUNCTION
        = 0x67, // Channel Identifier - 4 bytes
00251 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_SELECT_MEDIA_FUNCTION
        = 0x68, // UI Function Media - 1 byte
00252 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_SELECT_A_V_INPUT_FUNCTION
        = 0x69, // UI Function Select A/V Input - 1 byte
00253 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_SELECT_AUDIO_INPUT_FUNCTION
        = 0x6A, // UI Function Select Audio Input - 1 byte
00254 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_POWER_TOGGLE_FUNCTION
        = 0x6B,
00255 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_POWER_OFF_FUNCTION
        = 0x6C,
00256 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_POWER_ON_FUNCTION
        = 0x6D,
00257 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_F1_BLUE
        = 0x71,
00258 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_F2_RED
        = 0x72,
00259 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_F3_GREEN
        = 0x73,
00260 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_F4_YELLOW
        = 0x74,
00261 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_F5
        = 0x75,
00262 EMBER_AF_RF4CE_ZRC_ACTION_BANK_HDMI_CEC_ACTION_CODE_DATA
        = 0x76,
00263 };
00264
00265 #ifndef DOXYGEN_SHOULD_SKIP_THIS
00266 enum EmberAfRf4ceZrcActionBankHomeAutomationActionCode
00267 #else
00268 typedef uint8_t EmberAfRf4ceZrcActionBankHomeAutomationActionCode
00269 ;
00270 enum
00271 #endif
00272 {
00273
00274
00275
00276     EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_SCENES_STORE_LOCAL_SCENE_0
        = 0x00,
00277
00278     EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_SCENES_STORE_LOCAL_SCENE_1
        = 0x01,
00279
00280     EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_SCENES_STORE_LOCAL_SCENE_2
        = 0x02,
00281
00282     EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_SCENES_STORE_LOCAL_SCENE_3
        = 0x03,
00283
00284     EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_SCENES_STORE_LOCAL_SCENE_4
        = 0x04,
00285
00286     EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_SCENES_STORE_LOCAL_SCENE_5
        = 0x05,

```

```

00281
    EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_SCENES_STORE_LOCAL_SCENE_6
        = 0x06,
00282
    EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_SCENES_STORE_LOCAL_SCENE_7
        = 0x07,
00283
    EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_SCENES_STORE_LOCAL_SCENE_8
        = 0x08,
00284
    EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_SCENES_STORE_LOCAL_SCENE_9
        = 0x09,
00285
    EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_SCENES_STORE_LOCAL_SCENE_10
        = 0x0A,
00286
    EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_SCENES_STORE_LOCAL_SCENE_11
        = 0x0B,
00287
    EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_SCENES_STORE_LOCAL_SCENE_12
        = 0x0C,
00288
    EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_SCENES_STORE_LOCAL_SCENE_13
        = 0x0D,
00289
    EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_SCENES_STORE_LOCAL_SCENE_14
        = 0x0E,
00290
    EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_SCENES_STORE_LOCAL_SCENE_15
        = 0x0F,
00291
    EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_SCENES_RECALL_LOCAL_SCENE_0
        = 0x10,
00292
    EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_SCENES_RECALL_LOCAL_SCENE_1
        = 0x11,
00293
    EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_SCENES_RECALL_LOCAL_SCENE_2
        = 0x12,
00294
    EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_SCENES_RECALL_LOCAL_SCENE_3
        = 0x13,
00295
    EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_SCENES_RECALL_LOCAL_SCENE_4
        = 0x14,
00296
    EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_SCENES_RECALL_LOCAL_SCENE_5
        = 0x15,
00297
    EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_SCENES_RECALL_LOCAL_SCENE_6
        = 0x16,
00298
    EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_SCENES_RECALL_LOCAL_SCENE_7
        = 0x17,
00299
    EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_SCENES_RECALL_LOCAL_SCENE_8
        = 0x18,
00300
    EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_SCENES_RECALL_LOCAL_SCENE_9
        = 0x19,

```

```

00301
    EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_SCENES_RECALL_LOCAL_SCENE_10
        = 0x1A,
00302
    EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_SCENES_RECALL_LOCAL_SCENE_11
        = 0x1B,
00303
    EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_SCENES_RECALL_LOCAL_SCENE_12
        = 0x1C,
00304
    EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_SCENES_RECALL_LOCAL_SCENE_13
        = 0x1D,
00305
    EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_SCENES_RECALL_LOCAL_SCENE_14
        = 0x1E,
00306
    EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_SCENES_RECALL_LOCAL_SCENE_15
        = 0x1F,
00307    EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_ON_OFF_OFF
        = 0x20,
00308    EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_ON_OFF_ON
        = 0x21,
00309    EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_ON_OFF_TOGGLE
        = 0x22,
00310
    EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_LEVEL_CONTROL_MOVE_TO_LEVEL
        = 0x30,
00311    EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_LEVEL_CONTROL_MOVE
        = 0x31,
00312    EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_LEVEL_CONTROL_STEP
        = 0x32,
00313    EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_LEVEL_CONTROL_STOP
        = 0x33,
00314
    EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_LEVEL_CONTROL_MOVE_TO_LEVEL_WITH_ON_OFF
        = 0x34,
00315
    EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_LEVEL_CONTROL_MOVE_WITH_ON_OFF
        = 0x35,
00316
    EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_LEVEL_CONTROL_STEP_WITH_ON_OFF
        = 0x36,
00317
    EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_LEVEL_CONTROL_STOP_WITH_ON_OFF
        = 0x37,
00318    EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_DOOR_LOCK_LOCK_DOOR
        = 0x40,
00319
    EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_DOOR_LOCK_UNLOCK_DOOR
        = 0x41,
00320    EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_DOOR_LOCK_TOGGLE
        = 0x42,
00321
    EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_DOOR_LOCK_UNLOCK_WITH_TIMEOUT
        = 0x43,
00322
    EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_WINDOW_COVERING_UP_OPEN
        = 0x50,
00323
    EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_WINDOW_COVERING_DOWN_CLOSE
        = 0x51,
00324    EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_WINDOW_COVERING_STOP
        = 0x52,

```

```

00325
    EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_WINDOW_COVERING_GO_TO_LIFT_VALUE
        = 0x54,
00326
    EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_WINDOW_COVERING_GO_TO_LIFT_PERCENTAGE
        = 0x55,
00327
    EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_WINDOW_COVERING_GO_TO_TILT_VALUE
        = 0x57,
00328
    EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_WINDOW_COVERING_GO_TO_TILT_PERCENTAGE
        = 0x58,
00329
    EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_THERMOSTAT_SETPOINT_RAISE_LOWER
        = 0x60,
00330
    EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_COLOR_CONTROL_MOVE_TO_HUE
        = 0x70,
00331
    EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_COLOR_CONTROL_MOVE_HUE
        = 0x71,
00332
    EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_COLOR_CONTROL_STEP_HUE
        = 0x72,
00333
    EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_COLOR_CONTROL_MOVE_TO_SATURATION
        = 0x73,
00334
    EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_COLOR_CONTROL_MOVE_SATURATION
        = 0x74,
00335
    EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_COLOR_CONTROL_STEP_SATURATION
        = 0x75,
00336
    EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_COLOR_CONTROL_MOVE_TO_HUE_AND_SATURATION
        = 0x76,
00337
    EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_COLOR_CONTROL_MOVE_TO_COLOR
        = 0x77,
00338
    EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_COLOR_CONTROL_MOVE_COLOR
        = 0x78,
00339
    EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_COLOR_CONTROL_STEP_COLOR
        = 0x79,
00340
    EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_COLOR_CONTROL_MOVE_TO_COLOR_TEMPERATURE
        = 0x7A,
00341    EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_IAS_ACE_ARM
            = 0xC4,
00342    EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_IAS_ACE_BYPASS
            = 0xC5,
00343    EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_IAS_ACE_EMERGENCY
            = 0xC6,
00344    EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_IAS_ACE_FIRE
            = 0xC7,
00345    EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_IAS_ACE_PANIC
            = 0xC8,
00346
    EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_PREVIOUS_DESTINATION_GROUP
        = 0xF0,
00347

```

```

EMBER_AF_RF4CE_ZRC_ACTION_BANK_HOME_AUTOMATION_ACTION_CODE_NEXT_DESTINATION_GROUP
= 0xF1,
00348 };
00349
00353 typedef struct {
00354     uint8_t pairingIndex;
00355     EmberAfRf4ceZrcActionType actionType;
00356     EmberAfRf4ceZrcModifierBit modifierBits
    ;
00357     uint8_t actionPayloadLength;
00358     EmberAfRf4ceZrcActionBank actionBank;
00359     EmberAfRf4ceZrcActionCode actionCode;
00360     uint16_t actionVendorId;
00361     const uint8_t *actionPayload;
00362     uint16_t timeMs;
00363 } EmberAfRf4ceZrcActionRecord;
00364
00368 typedef struct
00369 {
00370     EmberAfRf4ceDeviceType actionDeviceType
    ;
00371     EmberAfRf4ceZrcActionBank actionBank;
00372     EmberAfRf4ceZrcActionCode actionCode;
00373 } EmberAfRf4ceZrcMappableAction;
00374
00378 #define EMBER_AF_RF4CE_ZRC_ACTION_MAPPING_MAPPING_FLAGS_RF_SPECIFIED_BIT
0x01
00379 #define EMBER_AF_RF4CE_ZRC_ACTION_MAPPING_MAPPING_FLAGS_IR_SPECIFIED_BIT
0x02
00380 #define EMBER_AF_RF4CE_ZRC_ACTION_MAPPING_MAPPING_FLAGS_RF_DESCRIPTOR_FIRST_BIT
0x04
00381 // Bits 3-5 are reserved.
00382 #define EMBER_AF_RF4CE_ZRC_ACTION_MAPPING_MAPPING_FLAGS_USE_DEFAULT_BIT
0x40
00383 #define EMBER_AF_RF4CE_ZRC_ACTION_MAPPING_MAPPING_FLAGS_PERMANENT_BIT
0x80
00384
00388 // Bits 0-3: minimum number of transmissions.
00389 #define RF4CE_ZRC_ACTION_MAPPING_RF_CONFIG_KEEP_TRANSMITTING_UNTIL_KEY_RELEASE
0x10
00390 #define RF4CE_ZRC_ACTION_MAPPING_RF_CONFIG_SHORT_RF_RETRY
0x20
00391 #define RF4CE_ZRC_ACTION_MAPPING_RF_CONFIG_ATOMIC_ACTION
0x40
00392 // Bit 7 is reserved.
00393
00397 #define RF4CE_ZRC_ACTION_MAPPING_IR_CONFIG_VENDOR_SPECIFIC 0x01
00398 // Bits 1-7 are reserved.
00399
00403 typedef struct {
00404     uint8_t mappingFlags;
00405
00406     uint8_t rfConfig;
00407     uint8_t rf4ceTxOptions;
00408     uint8_t actionDataLength;
00409     uint8_t* actionData;
00410
00411     uint8_t irConfig;
00412     uint16_t irVendorId;
00413     uint8_t irCodeLength;
00414     uint8_t* irCode;
00415 } EmberAfRf4ceZrcActionMapping;
00416
00420 typedef struct {
00421     uint8_t contents[32];
00422 } EmberAfRf4ceZrcHomeAutomationSupported;
00423
00427 typedef struct {
00428     uint8_t *contents;
00429     uint8_t contentsLength;
00430     uint8_t instanceId;
00431     uint8_t attributeId;
00432 } EmberAfRf4ceZrcHomeAutomationAttribute;
00433
00437 #define EMBER_AF_RF4CE_ZRC_COMMANDS_SUPPORTED_SIZE 32
00438
00442 typedef struct {
00444     uint8_t contents[EMBER_AF_RF4CE_ZRC_COMMANDS_SUPPORTED_SIZE
    ];
00445 } EmberAfRf4ceZrcCommandsSupported;

```

```
00446
00447 #endif // __RF4CE_ZRC20_TYPES_H__
```

8.78 rf4ce-zrc20.h File Reference

```
#include "rf4ce-zrc20-types.h"
```

Macros

- #define ACTION_MAPPING_CLIENT
- #define ACTION_MAPPING_SERVER
- #define SET_DEFAULT(entry)

Functions

- EmberStatus emberAfRf4ceZrc20Bind (EmberAfRf4ceDeviceType searchDevType)
- EmberStatus emberAfRf4ceZrc20ProxyBind (EmberPanId panId, EmberEUI64 ieeeAddr)
- EmberStatus emberAfRf4ceZrc20ActionStart (uint8_t pairingIndex, EmberAfRf4ceZrcActionBank actionBank, EmberAfRf4ceZrcActionCode actionCode, EmberAfRf4ceZrcModifierBit actionModifier, uint16_t actionVendorId, const uint8_t *actionData, uint8_t actionDataLength, bool atomic)
- EmberStatus emberAfRf4ceZrc20ActionStop (uint8_t pairingIndex, EmberAfRf4ceZrcActionBank actionBank, EmberAfRf4ceZrcActionCode actionCode, EmberAfRf4ceZrcModifierBit actionModifier, uint16_t actionVendorId)
- EmberStatus emberAfRf4ceZrc20LegacyCommandDiscovery (uint8_t pairingIndex)
- EmberStatus emberAfRf4ceZrc20StartActionMappingsNegotiation (uint8_t pairingIndex)
- EmberStatus emberAfRf4ceZrc20StartHomeAutomationSupportedAnnouncement (uint8_t pairingIndex)
- EmberStatus emberAfRf4ceZrc20PullHomeAutomationAttribute (uint8_t pairingIndex, uint16_t vendorId, uint8_t haInstanceId, uint8_t haAttributeId)
- uint8_t *emberAfRf4ceZrcCommandsSupportedContents (EmberAfRf4ceZrcCommandsSupported *commandsSupported)
- bool emberAfRf4ceZrc20ActionMappingEntryHasRfDescriptor (const EmberAfRf4ceZrcActionMapping *entry)
- bool emberAfRf4ceZrc20ActionMappingEntryHasIrDescriptor (const EmberAfRf4ceZrcActionMapping *entry)
- bool emberAfRf4ceZrc20ActionMappingEntryHasIrVendorId (const EmberAfRf4ceZrcActionMapping *entry)

8.79 rf4ce-zrc20.h

```
00001 // Copyright 2014 Silicon Laboratories, Inc.
00002
00003 #ifndef __RF4CE_ZRC20_H__
00004 #define __RF4CE_ZRC20_H__
00005
00006 #include "rf4ce-zrc20-types.h"
00007
00107 #ifdef EMBER_AF_RF4CE_ZRC_IS_ACTIONS_ORIGINATOR
00108
00109 #define EMBER_AF_RF4CE_ZRC_IS_ORIGINATOR
00110#endif
```

```

00111
00112 #ifdef EMBER_AF_RF4CE_ZRC_IS_ACTIONS_RECIPIENT
00113     #define EMBER_AF_RF4CE_ZRC_IS_RECIPIENT
00115 #endif
00116
00117 #define ACTION_MAPPING_CLIENT 1
00118 #define ACTION_MAPPING_SERVER 2
00119
00120 #if EMBER_AF_PLUGIN_RF4CE_ZRC20_ACTION_MAPPING_SUPPORT == ACTION_MAPPING_CLIENT
00121     #define EMBER_AF_PLUGIN_RF4CE_ZRC20_IS_ACTION_MAPPING_CLIENT
00123
00124 #if defined(EMBER_AF_RF4CE_ZRC_IRDB_VENDOR_IDS)
00125 #define EMBER_AF_PLUGIN_RF4CE_ZRC20_LOCAL_IRDB_VENDOR_ATTRIBUTE_SUPPORT
00126 #endif
00127
00128 #elif EMBER_AF_PLUGIN_RF4CE_ZRC20_ACTION_MAPPING_SUPPORT ==
00129     ACTION_MAPPING_SERVER
00130     #define EMBER_AF_PLUGIN_RF4CE_ZRC20_IS_ACTION_MAPPING_SERVER
00131
00132     #define EMBER_AF_PLUGIN_RF4CE_ZRC20_REMOTE_IRDB_VENDOR_ATTRIBUTES_SUPPORT
00134 #endif
00135
00136 EmberStatus emberAfRf4ceZrc20Bind(EmberAfRf4ceDeviceType
00137     searchDevType);
00138 EmberStatus emberAfRf4ceZrc20ProxyBind(EmberPanId
00139     panId,
00140             EmberEUI64 ieeeAddr);
00140
00175 EmberStatus emberAfRf4ceZrc20ActionStart(uint8_t
00176     pairingIndex,
00176             EmberAfRf4ceZrcActionBank
00177
00177     actionBank,
00178             EmberAfRf4ceZrcActionCode
00178
00178     actionCode,
00178             EmberAfRf4ceZrcModifierBit
00179
00179     actionModifier,
00180             uint16_t actionVendorId,
00180             const uint8_t *actionData,
00181             uint8_t actionDataLength,
00182             bool atomic);
00183
00204 EmberStatus emberAfRf4ceZrc20ActionStop(uint8_t
00205     pairingIndex,
00205             EmberAfRf4ceZrcActionBank
00206
00206     actionBank,
00207             EmberAfRf4ceZrcActionCode
00207
00207     actionCode,
00207             EmberAfRf4ceZrcModifierBit
00208
00208     actionModifier,
00208             uint16_t actionVendorId);
00225 EmberStatus emberAfRf4ceZrc20LegacyCommandDiscovery
00226     (uint8_t pairingIndex);
00237 EmberStatus emberAfRf4ceZrc20StartActionMappingsNegotiation
00238     (uint8_t pairingIndex);
00249 EmberStatus emberAfRf4ceZrc20StartHomeAutomationSupportedAnnouncement
00249     (uint8_t pairingIndex);
00250
00271 EmberStatus emberAfRf4ceZrc20PullHomeAutomationAttribute
00272     (uint8_t pairingIndex,
00273             uint16_t vendorId,
00274             uint8_t haInstanceId,
00274             uint8_t haAttributeId)
00275 ;
00275
00284 #ifdef DOXYGEN_SHOULD_SKIP_THIS
00285     uint8_t *emberAfRf4ceZrcCommandsSupportedContents
00285     (EmberAfRf4ceZrcCommandsSupported *
00285     commandsSupported);
00286 #else
00287     #define emberAfRf4ceZrcCommandsSupportedContents(commandsSupported) \
00288         ((commandsSupported)->contents)
00289 #endif
00290
00291 #ifdef DOXYGEN_SHOULD_SKIP_THIS

```

```

00292
00299     bool emberAfRf4ceZrc20ActionMappingEntryHasRfDescriptor
00300     (const EmberAfRf4ceZrcActionMapping *entry);
00301 #else
00302     #define emberAfRf4ceZrc20ActionMappingEntryHasRfDescriptor(entry)
00303         \
00304         (READBITS((entry)->mappingFlags,
00305             \
00306             (EMBER_AF_RF4CE_ZRC_ACTION_MAPPING_MAPPING_FLAGS_RF_SPECIFIED_BIT
00307             \
00308             | EMBER_AF_RF4CE_ZRC_ACTION_MAPPING_MAPPING_FLAGS_USE_DEFAULT_BIT)) \
00309             == EMBER_AF_RF4CE_ZRC_ACTION_MAPPING_MAPPING_FLAGS_RF_SPECIFIED_BIT)
00310 #endif
00311
00312 #ifdef DOXYGEN_SHOULD_SKIP_THIS
00313
00314     bool emberAfRf4ceZrc20ActionMappingEntryHasIrDescriptor
00315     (const EmberAfRf4ceZrcActionMapping *entry);
00316 #else
00317     #define emberAfRf4ceZrc20ActionMappingEntryHasIrDescriptor(entry)
00318         \
00319         (READBITS((entry)->mappingFlags,
00320             \
00321             (EMBER_AF_RF4CE_ZRC_ACTION_MAPPING_MAPPING_FLAGS_IR_SPECIFIED_BIT
00322             \
00323             | EMBER_AF_RF4CE_ZRC_ACTION_MAPPING_MAPPING_FLAGS_USE_DEFAULT_BIT)) \
00324             == EMBER_AF_RF4CE_ZRC_ACTION_MAPPING_MAPPING_FLAGS_IR_SPECIFIED_BIT)
00325 #endif
00326
00327 #ifdef DOXYGEN_SHOULD_SKIP_THIS
00328
00329     bool emberAfRf4ceZrc20ActionMappingEntryHasIrVendorId
00330     (const EmberAfRf4ceZrcActionMapping *entry);
00331 #else
00332     #define emberAfRf4ceZrc20ActionMappingEntryHasIrVendorId(entry) \
00333         (READBITS((entry)->irConfig,
00334             \
00335             RF4CE_ZRC_ACTION_MAPPING_IR_CONFIG_VENDOR_SPECIFIC) \
00336             == RF4CE_ZRC_ACTION_MAPPING_IR_CONFIG_VENDOR_SPECIFIC)
00337 #endif
00338
00339 // If the Use Default bit is set, the RF Specified and IR Specified bits and
00340 // their corresponding descriptors are ignored.
00341 #define SET_DEFAULT(entry)
00342     \
00343     ((entry)->mappingFlags =
00344         EMBER_AF_RF4CE_ZRC_ACTION_MAPPING_MAPPING_FLAGS_USE_DEFAULT_BIT)
00345 #endif // __RF4CE_ZRC20_H__
00346
00347 // END addtogroup
00348

```