ADLINK PacketManager
API Programming Guide

Manual Revision:	1.1
Revision Date:	August 15, 2017
Part Number:	50-1G043-1010

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000
Leading EDGE COMPUTING

Preface

Copyright 2016, 2017 ADLINK Technology, Inc.

This document contains proprietary information protected by copyright. All rights are reserved. No part of this manual may be reproduced by any mechanical, electronic, or other means in any form without prior written permission of the manufacturer.

Disclaimer

The information in this document is subject to change without prior notice in order to improve reliability, design, and function and does not represent a commitment on the part of the manufacturer. In no event will the manufacturer be liable for direct, indirect, special, incidental, or consequential damages arising out of the use or inability to use the product or documentation, even if advised of the possibility of such damages.

Environmental Responsibility

ADLINK is committed to fulfill its social responsibility to global environmental preservation through compliance with the European Union's Restriction of Hazardous Substances (RoHS) directive and Waste Electrical and Electronic Equipment (WEEE) directive. Environmental protection is a top priority for ADLINK. We have enforced measures to ensure that our products, manufacturing processes, components, and raw materials have as little impact on the environment as possible. When products are at their end of life, our customers are encouraged to dispose of them in accordance with the product disposal and/or recovery programs prescribed by their nation or company.

Trademarks

Product names mentioned herein are used for identification purposes only and may be trademarks and/or registered trademarks of their respective companies.

Revision History

Rev	Date	Description
1.00	22/03/2016	Initial release
1.1	15/08/2017	Separate the API sections for switch and processor blades respectively. Add VRRP configuration note. Update for MXN-3610/MXN-4100. Add LBG and CMM configuration note. Remove API for processor blades.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Table of Contents

Preface **2**
1 Introduction **5**
1.1 Architecture Overview 5
1.2 Deliverables 6
2 Application API **7**
2.1 Base 7
2.1.1 Macro Definitions 7
2.1.2 Data Structure 8
2.1.3 Enum Type 8
2.1.4 Functions 11
2.2 Static Routing (IPv4) 15
2.2.1 Macro Definitions 15
2.2.2 Enum Type 16
2.2.3 Data Structure 16
2.2.4 Functions 17
2.3 Static Routing (IPv6) 20
2.3.1 Macro Definitions 20
2.3.2 Enum Type 21
2.3.3 Data Structure 21
2.3.4 Functions 22
2.4 Port 25
2.4.1 Macro Definitions 25
2.4.2 Enum Type 26
2.4.3 Data Structure 29
2.4.4 Function 34
2.5 VLAN 38
2.5.1 Macro Definitions 38
2.5.2 Enum Type 38
2.5.3 Data Structure 39
2.5.4 Functions 40
2.6 Trunk 44
2.6.1 Macro Definitions 44
2.6.2 Enum Type 45
2.6.3 Data Structure 48
2.6.4 Functions 51
2.7 LBG 64
2.7.1 Macro Definitions 64
2.7.2 EnumType 66

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.7.3 Data Structure 66
2.7.4 Functions 66
2.8 Layer-2 70
2.8.1 Macro Definitions 70
2.8.2 Enum Type 70
2.8.3 Data Structure 70
2.8.4 Functions 71
2.9 Access Control List 74
2.9.1 Macro Definitions 74
2.9.2 Enum Type 74
2.9.3 Data Structure 77
2.9.4 Functions 79
2.10 Spanning Tree Protocol 82
2.10.1 Macro Definitions 82
2.10.2 Enum Type 82
2.10.3 Data Structure 83
2.10.4 Functions 89
2.11 Rapid Spanning Tree Protocol 96
2.11.1 Macro Definitions 96
2.11.2 Enum Type 96
2.11.3 Data Structure 99
2.11.4 Functions 109
2.12 LLDP 116
2.12.1 Macro Definitions 116
2.12.2 Enum Type 117
2.12.3 Data Structure 118
2.12.4 Functions 122
2.13 VRRP 127
2.13.1 Macro Definitions 127
2.13.2 EnumType 127
2.13.3 Data Structure 128
2.13.4 Functions 129
2.14 CMM 132
2.14.1 Macro Definitions 132
2.14.2 EnumType 132
2.14.3 Data Structure 132
2.14.4 Functions 136
Getting Service 141

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

1 Introduction

This guide describes how use the ADLINK PacketManager Software application program interface (API).

1.1 Architecture Overview

All the APIs listed in this document are programmed using the hardware architecture, shown in Figure 1. In this figure, the client is running on a standalone server. However, it also can run on the Local Management Processor (LMP) on the blade.

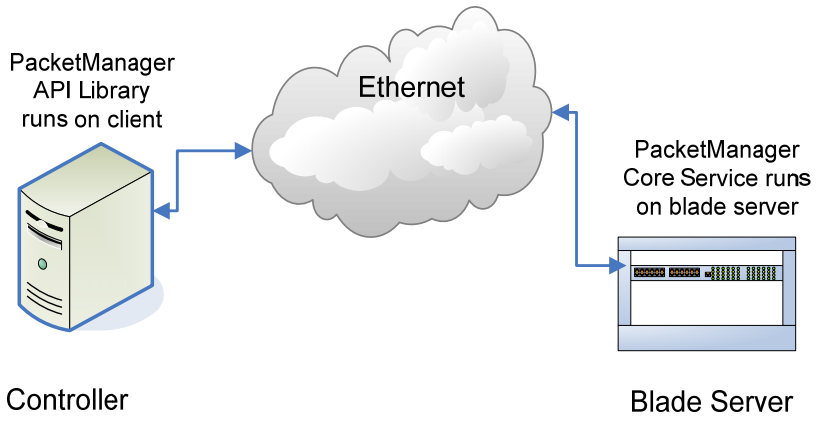


Figure 1 Hardware Architecture for PacketManager API Programming

ADLINK PacketManager Software implements the network connections between server and client so that customers do not have to worry about the connections between them. Client API libraries, header files and relevant documentation are provided for easy programming.

PacketManager is designed to support different hardware architectures, including Broadcom and Intel® Red Rock Canyon switches. PacketManager provides application APIs to handle operations for protocol stack, ports and other modules. Some APIs are dedicated for specific switch boards and others are general purpose.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

1.2 Deliverables

The ADLINK PacketManager API Library includes the following software components:

Include:	Header files of the ADLINK PackageManager API library
Lib:	Static library including the API functions listed in this document
Bin:	Binary files for controlling the system

Example: Application demo to show usage of the APIs provided

ReadMe:	Release notes for different library revisions

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2 Application API

2.1 Base

This section describes the macro, enum, struct and functions for fundamentals. They are defined in the header file “adlink_api_base.h” or “adlink_base.h”. Users do not need to worry about where they are defined. Just include the header file “adlink_api_base.h”.

2.1.1 Macro Definitions

MAC_SIZE

MAC address size

Supported On

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

PROTOCOL_UDP

UDP protocol id.

Supported On

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

PROTOCOL_TCP

TCP protocol id

Supported On

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

PROTOCOL_ICMP

ICMP protocol id

Supported On

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

ADLINK_EVENT_SWITCH_DOWN

Switch blade down event (RPC lost).

Supported On

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.1.2 Data Structure

adlink_ether_address_t

```
typedef struct adlink_ether_address_s
{
    uint8_t addr_bytes[ETHER_ADDR_LENGTH]; /**< Address bytes in transmission
order */
} __attribute__ ((__packed__)) adlink_ether_address_t;
```

Define the Ethernet address format.

Supported On

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

2.1.3 Enum Type

status_t

```
typedef enum
{
    ADLINK_E_None = 0,
    ADLINK_E_INTERNAL = -1,
    ADLINK_E_MALLOC = -2,
    ADLINK_E_PARA = -3,
    ADLINK_E_TABLE_EMPTY = -4,
    ADLINK_E_TABLE_FULL = -5,
    ADLINK_E_NOT_FOUND = -6,
    ADLINK_E_FIND = ADLINK_E_NOT_FOUND,
    ADLINK_E_EXIST = -7,
    ADLINK_E_TIMEOUT = -8 ,
    ADLINK_E_BUSY = -9,
    ADLINK_E_FAIL = -10,
    ADLINK_E_DISABLED = -11,
    ADLINK_E_BADID = -12,
    ADLINK_E_RESOURCE = -13,
    ADLINK_E_CONFIG = -14,
    ADLINK_E_UNAVAIL = -15,
    ADLINK_E_INIT = -16,
    ADLINK_E_SOCKET = -17,
    ADLINK_E_RING = -18,
    ADLINK_E_IPC = -19,
    ADLINK_E_RPC = -20,
    ADLINK_E_FILE = -21,
```


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
        ADLINK_E_RANGE = -22,
        ADLINK_E_NO_LOCK = -23,
        ADLINK_E_AUTH = -24,
        ADLINK_E_NOMSG = -25,
        ADLINK_E_OVERFLOW = -26,
        ADLINK_E_NOT_SUPPORT = -27,
        ADLINK_E_BAD_MSG = -28,
        ADLINK_E_FUNC_NOT_IMPLMENT = -29,
        ADLINK_E_NOENT = -30,
        ADLINK_E_NOSPC = -31,
        ADLINK_E_NO_TAILQ = -32,
        ADLINK_E_OPEN = -33,
        ADLINK_E_ROUTE = -34,
        ADLINK_E_PTHREAD = -35,
        ADLINK_E_PORT_EXIST   = -36,
        ADLINK_E_PORT_FIND = -37,
        ADLINK_E_OTHER = -127,
        ADLINK_E_SERVER_BASE = -128,
    } status_t;
```

Supported On

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

This enum describes the return codes of the APIs. They are defined in “adlink_api_base.h”. Table 1 shows the meaning of each error code.

Return Code	Description
ADLINK_E_None	The routine completed successfully
ADLINK_E_INTERNAL	Some internal operations failed in an unexpected way. This usually indicates that an access to some hardware resource failed
ADLINK_E_MALLOC	Dynamically allocated memory was not available to complete the operation
ADLINK_E_PARA	A parameter to a ADLINK API routines was invalid
ADLINK_E_TABLE_EMPTY	The routine expected a table not to be empty
ADLINK_E_TABLE_FULL	The routine could not add an entry to the table
ADLINK_E_NOT_FOUND ADLINK_E_FIND	The specific entry in a table could not be found
ADLINK_E_EXIST	The specific entry was already in a table
ADLINK_E_TIMEOUT	The routine started an operation that did not complete
ADLINK_E_BUSY	The routine tried to start an operation, but a previous operation was still running
ADLINK_E_FAIL	The routine attempted an operation that could not be

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Return Code	Description
	completed. This may be caused by a hardware failure or a configuration failure
ADLINK_E_DISABLED	The routine could not complete because a required feature was temporarily disabled
ADLINK_E_BADID	An identifier was not valid
ADLINK_E_RESOURCE	The routine needed more resources
ADLINK_E_CONFIG	Invalid configuration
ADLINK_E_UNAVAIL	Feature unavailable
ADLINK_E_INIT	Feature not initialized
ADLINK_E_SOCKET	Socket error
ADLINK_E_RING	RING error
ADLINK_E_IPC	IPC error
ADLINK_E_RPC	RPC error
ADLINK_E_FILE	File operate error
ADLINK_E_RANGE	Result too large or too small
ADLINK_E_NO_LOCK	No locks available
ADLINK_E_AUTH	Authentication error
ADLINK_E_NOMSG	No message of desired type
ADLINK_E_OVERFLOW	Value too large to be stored in data type
ADLINK_E_NOT_SUPPORTED	Not supported
ADLINK_E_BAD_MSG	Bad or Corrupt message
ADLINK_E_FUNC_NOT_IMPLMENT	Function not implemented
ADLINK_E_NOENT	No such file or such directory
ADLINK_E_NOSPC	No space left on device
ADLINK_E_NO_TAILQ	Uninitialized TAIL Queue
ADLINK_E_OPEN	Open/fopen operation failed
ADLINK_E_ROUTE	Route error
ADLINK_E_PTHREAD	Pthread error
ADLINK_E_PORT_EXIST	Port already exist
ADLINK_E_PORT_FIND	Port not find
ADLINK_E_OTHER	Error except above
ADLINK_E_SERVER_BASE	For a meaningful error code form server, we return it with this value added. So, if the returned code is less than this value, we can subtract this value and get the real return value from server. This is useful for debugging.

Table 1. Return Code Description

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.1.4 Functions

adlink_switch_chip_init ()

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Prototype

CLIENT* adlink_switch_chip_init(char* server_ip)

Description

This function is used to init the switch chip. It will return an RPC handle.

Parameters

server_ip [IN] IP address of the server

Return

CLIENT* the RPC client pointer, which will be used in all RPC api call.

adlink_switch_chip_close ()

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Prototype

status_t adlink_switch_chip_close(CLIENT* clnt)；

Description

This function is used to close the switch chip. It will close the RPC handle.

Parameters

clnt [IN] RPC client that the init function returned.

Return

status_t.

adlink_config_write ()

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Prototype

status_t adlink_config_write(CLIENT* clnt)

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Description
This function is used to write the current configuration into file. After the function is called successfully, the configuration will still be in effect after the system reboots.

Parameters
clnt [IN] RPC client that the init function returned.

Return
status_t

adlink_config_restore ()

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Prototype
status_t adlink_config_restore (CLIENT* clnt)

Description
This function is used to clear the configuration file. After the function is called successfully, the configuration will be cleared. This function doesn't change the current configuration of the system.

Parameters
clnt [IN] RPC client that the init function returned.

Return
status_t

adlink_switch_status_notify_register ()

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Prototype
status_t adlink_switch_status_notify_register(CLIENT* clnt,switch_status_callback_fun func);

Description
This function is used to regist a callback function. When the RPC connect with the blade is lost. The callback function will be called automaticly.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Parameters

clnt [IN] RPC client that the init function returned.

Func [IN] Call back function.

Return

status_t

adlink_get_board_type

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Prototype

char* adlink_get_board_type(void)

Description

This function is used to get the board type

Parameters

None.

Return

Board type string.

adlink_fabric_unit_id_get ()

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Prototype

int adlink_fabric_unit_id_get(CLIENT* clnt)

Description

This function is used to get the fabric unit id.

Parameters

clnt [IN] RPC client that the init function returned.

Return

Int Fabric unit id.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

adlink_base_unit_id_get ()

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Prototype

int adlink_base_unit_id_get(CLIENT* clnt)

Description

This function is used to Base unit id.

Parameters

clnt [IN] RPC client that the init function returned.

Return

Int Fabric unit id.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.2 Static Routing (IPv4)

This section describes the macro, enum, struct and functions for routing. They are defined in the header file “adlink_api_route.h”.

2.2.1 Macro Definitions

MAX_REPEAT_ROUTE_NUM

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

This macro is defined the max num of the route entris with the same destination address.

MAX_METRIC_NUM

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

This macro is defined for the max metric.

TAPNAMSIZ

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

This macro is defined for the max length of the tap device name.

ADLINK_ROUTE_ENTRY_STATUS_KERNEL

ADLINK_ROUTE_ENTRY_STATUS_CHIP

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

This macro is defined for route entries status.kernel mean that the entry is working in linux kermel. Chip mean that the entry is working in the switch chip T-cam.

MAX_ROUTE_ENTRY_ONE_TIME_GET

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

This macro is used for getting route entries from the server. Due to packet size limitations allowed in the network (MTU), it isn't possible to get all the entries at once. This macro defines the maximum number of entries retrievable at one time.

2.2.2 Enum Type

adlink_route_type_t

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

```
typedef enum adlink_route_type
{
    ADLINK_ROUTE_TYPE_UNKNOWN = 0,
    ADLINK_ROUTE_TYPE_STATIC = 1,
    ADLINK_ROUTE_TYPE_DIRECT,
    ADLINK_ROUTE_TYPE_PROTOCOL,
}adlink_route_type_t;
```

Route type description.

Fields	Description
ADLINK_ROUTE_TYPE_UNKNOWN	Unknow route entry type
ADLINK_ROUTE_TYPE_STATIC	Static route entry type
ADLINK_ROUTE_TYPE_DIRECT	Direct route entry type
ADLINK_ROUTE_TYPE_PROTOCOL	Route entry generated by protocols like RIP,OPSF and so on

2.2.3 Data Structure

adlink_route_info_t

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

```
typedef struct adlink_route_info
{   int unit;
    UINT32 ip_addr;
    UINT32 ip_mask;
    UINT32 gateway_ip;
    UINT16 metric;
    UINT16 type;
```


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
    char    if_name[TAPNAMSIZ];
}adlink_route_info_t;
```

Fields	Description
ip_addr	IP address
ip_mask	IP address net mask
Gateway_ip	Next hop IP address
metric	metric
type	adlink_route_type_t
If_name	Dev name

2.2.4 Functions

adlink_route_add ()

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Prototype

status_t adlink_route_add(CLIENT* clnt, int unit,UINT32 ip_addr,UINT32 ip_mask,UINT32 gateway_ip,UINT16 metric);

Description

This function is used to add a rule entry.

Parameters

clnt [IN] RPC client that the init function returned.
uinit [IN] Fabric or Base Uinit.
ip_addr [IN] ip address.
ip_mask [IN] ip address mask
gateway_ip [IN] gateway ip
metric [IN] route metric

Return

status_t

adlink_route_delete ()

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Prototype

status_t adlink_route_delete(CLIENT* clnt, int unit,UINT32 ip_addr,UINT32 ip_mask,UINT32

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

gateway_ip,UINT16 metric);

Description

This function is used to delete a route entry.

Parameters

clnt [IN] RPC client that the init function returned.
uinit [IN] Fabric or Base Uinit.
ip_addr [IN] ip address.
ip_mask [IN] ip address mask.
gateway_ip [IN] gateway ip
metric [IN] route metric

Return

status_t.

adlink_route_table_clear ()

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Prototype

status_t adlink_route_table_clear(CLIENT* clnt,int unit);

Description

This function is used to clear all routing table entries that have been added by users.

Parameters

clnt [IN] RPC client that the init function returned.
uinit [IN] Fabric or Base Uinit.

Return

status_t.

adlink_route_table_dump ()

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Prototype

status_t adlink_route_table_dump (CLIENT* clnt,int unit,adlink_route_info_t* last_route,int *count,adlink_route_info_t*route_info);

Description

This function is used to get all route entry info that has been added.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Parameters

clnt	[IN] RPC client that the init function returned.
uinit	[IN] Fabric or Base Uinit.
ip_last /mask_last	[IN] These two parameters pass the last route entry that was retrieved. The function call will start retrieving route entry info after these values. Set both of these parameters to 0 the first time this function is called.
Count	[OUT] This parameter passes the number of route entry values retreived. If "Count" is less than MAX_ROUTE_ENTRY_ONE_TIME_GET, then all the route entries have been retrieved. Otherwise, we need to call this function again to continue retrieving route entry info.
route_info	[OUT] This parameter is the pointer for saving the route entry info. It must have enough space to save as many as MAX_ROUTE_ENTRY_ONE_TIME_GET route entries.

Return

status_t.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.3 Static Routing (IPv6)

This section describes the macro, enum, struct and functions for routing. They are defined in the header file “adlink_api_route.h”.

2.3.1 Macro Definitions

MAX_REPEAT_ROUTE_NUM

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

This macro is defined the max num of the route entris with the same destination address.

MAX_METRIC_NUM

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

This macro is defined for the max metric.

TAPNAMSIZ

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

This macro is defined for the max length of the tap device name.

MAX_ROUTE_ENTRY_ONE_TIME_GET

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

This macro is used for getting route entries from the server. Due to packet size limitations allowed in the network (MTU), it isn't possible to get all the entries at once. This macro defines the maximum number of entries retrievable at one time.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.3.2 Enum Type

adlink_route_type_t

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

```
typedef enum adlink_route_type
{
    ADLINK_ROUTE_TYPE_UNKNOWN = 0,
    ADLINK_ROUTE_TYPE_STATIC = 1,
    ADLINK_ROUTE_TYPE_DIRECT,
    ADLINK_ROUTE_TYPE_PROTOCOL,
}adlink_route_type_t;
```

Route type description.

Fields	Description
ADLINK_ROUTE_TYPE_UNKNOWN	Unknow route entry type
ADLINK_ROUTE_TYPE_STATIC	Static route entry type
ADLINK_ROUTE_TYPE_DIRECT	Direct route entry type
ADLINK_ROUTE_TYPE_PROTOCOL	Route entry generated by protocols like RIP,OPSF and so on

2.3.3 Data Structure

adlink_route6_info_t

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

```
typedef struct adlink_route6_info
{
    int unit;
    struct in6_addr ip6;
    UINT32 mask_len;
    struct in6_addr gateway_ip6;
    UINT16 metric;
    UINT16 type;
    char    if_name[TAPNAMSIZ];
}adlink_route6_info_t;
```


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Fields	Description
Ip6	IPv6 address
mask_len	IPv6 mask leng
gateway_ip6	Next hop IPv6 address
metric	Metric
type	adlink_route_type_t
if_name	Dev name

2.3.4 Functions

adlink_route6_add ()

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Prototype

status_t adlink_route6_add(CLIENT* clnt,int unit,struct in6_addr *ip6,UINT32 mask_len,struct in6_addr *gateway_ip6,UINT16 metric);;

Description

This function is used to add a IPv6 rule entry.

Parameters

clnt [IN] RPC client that the init function returned.
uinit [IN] Fabric or Base Uinit.
Ip6_addr [IN] ipv6 address.
Mask_len [IN] mask length.
gateway_ip6 [IN] gateway ipv6 address.
metric [IN] route metric.

Return

status_t

adlink_route6_delete ()

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Prototype

status_t adlink_route6_delete(CLIENT* clnt,int unit,struct in6_addr *ip6,UINT32 mask_len,struct in6_addr *gateway_ip6,UINT16 metric);

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Description

This function is used to delete a route entry.

Parameters

clnt [IN] RPC client that the init function returned.
uinit [IN] Fabric or Base Uinit.
Ip6_addr [IN] ipv6 address.
Mask_len [IN] mask length.
gateway_ip6 [IN] gateway ipv6 address.
metric [IN] route metric.

Return

status_t.

adlink_route6_table_clear ()

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Prototype

status_t adlink_route6_table_clear(CLIENT* clnt,int unit);

Description

This function is used to clear all routing table entries that have been added by users.

Parameters

clnt [IN] RPC client that the init function returned.
uinit [IN] Fabric or Base Uinit.

Return

status_t.

adlink_route_table_dump ()

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Prototype

status_t adlink_route6_table_dump(CLIENT* clnt,int unit,adlink_route6_info_t* last_route6,int *count,adlink_route6_info_t*route6_info);

Description

This function is used to get all route entry info that has been added.

Parameters

clnt	[IN] RPC client that the init function returned.
uinit	[IN] Fabric or Base Uinit.
last_route6	[IN] These parameter pass the last route entry that was retrieved. The function call will start retrieving route entry info after these values. Set all of these parameters to 0 the first time this function is called.
Count	[OUT] This parameter passes the number of route entry values retreived. If "Count" is less than MAX_ROUTE_ENTRY_ONE_TIME_GET, then all the route entries have been retrieved. Otherwise, we need to call this function again to continue retrieving route entry info.
Route6_info	[OUT] This parameter is the pointer for saving the route entry info. It must have enough space to save as many as MAX_ROUTE_ENTRY_ONE_TIME_GET route entries.

Return

status_t.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.4 Port

This section describes the macro, enum, struct and functions for port. They are defined in the header file “adlink_api_port.h”.

2.4.1 Macro Definitions

MAX_PORT_BCM_NAME_SIZE

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

This macro is defined the max length of port name string.

MAX_PORT_NUN

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

This macro is defined the max num of the ports of the blade.

TAPNAMSIZ

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

This macro is defined the max length of tap dev name.

ADLINK_PORT_LOOPBACK_OFF

ADLINK_PORT_LOOPBACK_TX2RX

ADLINK_PORT_LOOPBACK_RX2TX

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

This macro is defined the loopback mode of port.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.4.2 Enum Type

port_speed_t

```
typedef enum port_speed_e
    {
        PORT_SPEED_10M    = 10,
        PORT_SPEED_100M = 100,
        PORT_SPEED_1G     = 1000,
        PORT_SPEED_10G    = 10000,
        PORT_SPEED_40G = 40000,
    } port_speed_t;
```

Supported On

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Define the codes of port speed mode.

port_attribute_t

```
typedef enum port_attribute_e
{
    PORT_ATTR_MASK_NONE        =    0,
    PORT_ATTR_MASK_STATUS     = (1U << 0),
    PORT_ATTR_MASK_AUTONEG    = (1U << 1),
    PORT_ATTR_MASK_SPEED      = (1U << 2),
    PORT_ATTR_MASK_PVID       = (1U << 3),
    PORT_ATTR_MASK_COS        = (1U << 4),
    PORT_ATTR_MASK_MTU        = (1U << 5),
    PORT_ATTR_MASK_QINQ       = (1U << 6),
    PORT_ATTR_MASK_MAC_LEARN        = (1U << 7),
    PORT_ATTR_MASK_VLAN_FILTER      = (1U << 8),
    PORT_ATTR_MASK_RATE_BCAST     = (1U << 9),
    PORT_ATTR_MASK_RATE_MCAST     = (1U << 10),
    PORT_ATTR_MASK_RATE_DLF       = (1U << 11),
    PORT_ATTR_MASK_TX_PAUSE       = (1U << 12),
    PORT_ATTR_MASK_RX_PAUSE       = (1U << 13),
    PORT_ATTR_MASK_LOOPBACK       = (1u << 14),
}port_attribute_t;
```

Supported On

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Define the opcodes of port attributes.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

port_duplex_t

```
typedef enum port_duplex_e
{
    PORT_DUPLEX_HALF    = 0,
    PORT_DUPLEX_FULL    = 1,
}port_duplex_t;
```

Supported On

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Define the codes of port duplex. This attribute can be read only.

port_mirror_mode_t

```
typedef enum port_mirror_mode_e
{
    PORT_MIRROR_INGRESS     = (1U << 0),
    PORT_MIRROR_EGRESS      = (1U << 1),
}port_mirror_mode_t;
```

Supported On

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Define the codes of port mirror mode.

qinq_mode_t

```
typedef enum port_qinq_mode_e
{
    PORT_QINQ_MODE_NONE     = 0,
    PORT_QINQ_MODE_INTERNAL     = 1,
    PORT_QINQ_MODE_EXTERNAL     = 2,
}qinq_mode_t;
```

Supported On

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Define the codes of port qiniq mode.

port_link_status_t

```
typedef enum port_link_status_e
{ /*Heritage*/
```


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
    PORT_LINK_STATUS_DOWN          = 0,
    PORT_LINK_STATUS_UP            = 1,
    PORT_LINK_STATUS_FAILED        = 2,
    PORT_LINK_STATUS_REMOTE_FAULT  = 3,
    PORT_LINK_STATUS_LOCAL_FAULT   = 4,
} port_link_status_t;
```

Supported On

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Define the codes of port physical link status.

port_stp_state_t

```
typedef enum port_stp_state_e { /*Heritage */
    PORT_STP_DISABLE = 0, /* Disabled. */
    PORT_STP_BLOCK = 1, /* BPDUs/no learns. */
    PORT_STP_LISTEN = 2, /* BPDUs/no learns. */
    PORT_STP_LEARN = 3, /* BPDUs/learns. */
    PORT_STP_FORWARD = 4, /* Normal operation. */
    PORT_STP_COUNT = 5
} port_stp_state_t;
```

Supported On

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Define the codes of port status for Spanning Tree Protocol.

port_vlan_filter_mode_mask_t

```
typedef enum port_vlan_filter_mode_mask_e { /*Heritage*/
    PORT_VLAN_FILTER_INGRESS    = 0x00000001,
    PORT_VLAN_FILTER_EGRESS     = 0x00000002,
} port_vlan_filter_mode_mask_t;
```

Supported On

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Define the codes of VLAN filter mode in port attribute.

port_rate_limit_mask_t

```
typedef enum port_rate_limit_mask_e { /*Heritage*/
    PORT_RATE_DLF                  = 0x01,
    PORT_RATE_MCAST                = 0x02,
```


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
    PORT_RATE_BCAST          = 0x04,
    PORT_RATE_UCAST          = 0x08,
} port_rate_limit_mask_t;
```

Supported On

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Define the codes of rate limit type mask in port attribute.

nic_config_mask_t

```
typedef enum nic_config_mask
{
    NIC_CONFIG_IP_MASK    = (1U << 0), /*Set this bit to flag IP&MASK operation*/
    NIC_CONFIG_HWADDR = (1U << 1),
    NIC_CONFIG_STATUS = (1U << 2), /*Set this bit to flag NIC running status operation*/
}nic_config_mask_t;
```

Supported On

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Define the codes of NIC operation.

port_info_change_event_t

```
typedef enum port_info_change_event_e
{
    PORT_EVENT_NONE        =    0,
    PORT_LINKSTATUS_EVENT     = (1U << 0),
    PORT_SPEED_EVENT       = (1U << 1),
    PORT_MTU_EVENT         = (1U << 2),
}port_info_change_event_t;
```

Supported On

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Define the codes of port status change event.

2.4.3 Data Structure

adlink_port_info_t

```
typedef struct adlink_port_info_s
{
    char port_name[MAX_PORT_BCM_NAME_SIZE];
```


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
        int port_id;
        int enable;
        int linkstatus;
        int autoneg;
        int duplex;
        int speed;
        int cos;
        UINT16 pvid;
        adlink_port_vlan_t    port_vlan_info;
        int stp_state;
        int loopback;
        int pause_tx;
        int pause_rx;
        adlink_port_rate_t rate_bcast;
        adlink_port_rate_t rate_mcast;
        adlink_port_rate_t rate_dlf;
        int mtu;
        UINT32 fault;     /*link fault reason*/
        int qinq;
        int vlanfilter;
        int mac_learn;
}adlink_port_info_t;
```

Supported On

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Define the entry format of the port attribute table.

adlink_port_status_set_cmd_t

```
typedef struct adlink_port_status_set_cmd {
    int unit;
    UINT64 port_mask;             /*port mask*/
    UINT64 action_mask;           /*PORT_ATTR_xxx in port_attribute_t*/
    int status;             /*ENABLE/DISABLE*/
    int autoneg;     /*ENABLE/DISABLE*/
    int duplex;      /*PORT_DUPLEX_xxx in port_duplex_t*/
    int speed;       /*PORT_SPEED_xxx in port_speed_t*/
    UINT16 pvid;                  /*default 802.1Q vid for untagged packets*/
    int cos;                /*default 802.1Q cos for untagged packets*/
    int qinq;
    int mac_learn;
    int loopback;
    int pause_tx;
```


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
    int pause_rx;
    int mtu;
    int vlanfilter;
    adlink_port_rate_t rate_bcast;
    adlink_port_rate_t rate_mcast;
    adlink_port_rate_t rate_dlf;
}adlink_port_status_set_cmd_t;
```

Supported On

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Define the cmd struct of ports attribute setting.

adlink_port_status_get_cmd_t

```
typedef struct adlink_port_status_get_cmd
{
    int unit;
    UINT32 port;
}adlink_port_status_get_cmd_t;
```

Supported On

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Define the entry format of the cmd of port status geting.

adlink_mirror_t

```
typedef struct adlink_mirror_s {
    UINT32 enable;          /*ENABLE/DISABLE. Other items are valid only when 'enable' is
true.*/
    UINT32 mode;            /*PORT_MIRROR_xxx. Control mirror mode.*/
    UINT32 dst_port;       /*port of mirror to.*/
}adlink_mirror_t;
```

Supported On

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Define the entry format of the information of port mirror.

adlink_portstat_t

```
    typedef struct adlink_portstat_s
    {
        UINT64 rx_byte_num;
        UINT64 rx_pkt_num;
```


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
        UINT64 rx_pkt_unicast;
        UINT64 rx_pkt_err;
        UINT64 rx_pkt_discards;
        UINT64 rx_pkt_broadcast;
        UINT64 rx_pkt_multicast;
        UINT64 rx_pkt_octets64;
        UINT64 rx_pkt_octets65to127;
        UINT64 rx_pkt_octets128to255;
        UINT64 rx_pkt_octets256to511;
        UINT64 rx_pkt_octets512to1023;
        UINT64 rx_pkt_octets1024to1518;
        UINT64 rx_pkt_oversize;
        UINT64 rx_pkt_total;

        UINT64 tx_byte_num;
        UINT64 tx_pkt_num;
        UINT64 tx_pkt_unicast;
        UINT64 tx_pkt_discards;
        UINT64 tx_pkt_broadcast;
        UINT64 tx_pkt_multicast;
        UINT64 tm_use_last_clr; //us
    }adlink_portstat_t;
```

Supported On

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Define the entry format of the port statistic table.

adlink_set_port_status_cmd_t

```
    typedef struct adlink_set_port_status_cmd {
        UINT32 port;            /*port ID*/
        UINT32 flag;            /*PORT_ATTR_xxx in port_attribute_t*/
        int status;            /*ENABLE/DISABLE*/
        int autoneg_mode;      /*ENABLE/DISABLE*/
        int duplex_mode;       /*PORT_DUPLEX_xxx in port_duplex_t*/
        int speed_mode;        /*PORT_SPEED_xxx in port_speed_t*/
        int pvid;              /*default 802.1Q vid for untagged packets*/
        int cos;               /*default 802.1Q cos for untagged packets*/
        int mtu;               /*max data transmission unit*/
    } adlink_set_port_status_cmd_t;
```


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Supported On

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Define the format of parameters used to set port attributes.

adlink_nic_config_set_cmd_t

```
typedef struct adl_nic_config_set_cmd {
    int unit;
    char if_name[TAPNAMSIZ];
    UINT32 config_mask;          //nic_config_mask_t
    UINT32 ip;
    UINT32 netmask;
    UINT8   mac[MAC_SIZE];
    UINT8   rsv[2];
    UINT32 status;   //0 down, 1 up
}adlink_nic_config_set_cmd_t;
```

Supported On

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Define the format of parameters used to set NIC configuration.

adlink_nic_config_info_t

```
typedef struct adlink_nic_config_info
{
    UINT32 ip;
    UINT32 mask;
    UINT8   mac[MAC_SIZE];
    UINT8   rsv[2];
    UINT32 status;
}adlink_nic_config_info_t;
```

Supported On

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Define the format of retrieved NIC info.

adlink_nic_config_get_cmd_t

```
typedef struct adlink_nic_config_get_cmd {
    char if_name[TAPNAMSIZ];
}adlink_nic_config_get_cmd_t;
```


LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Supported On

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Define the format of the get NIC info command parameter.

2.4.4 Function

adlink_port_info_get

Prototype

status_t adlink_port_info_get(CLIENT* clnt ,uint8_t port, adlink_port_info_t* port_info);

Supported On

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Description

Gets port info.

Parameters

clnt	[IN] RPC client that the init function returned.
port	[IN] logic port id;
port_info	[OUT] port status info.

Return

status_t.

adlink_port_statistic_get

Prototype

status_t adlink_port_statistic_get (CLIENT* clnt, UINT32 port, adlink_portstat_t *stat);

Supported On

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Description

It’s to get port statistic info.

Parameters

clnt	[IN] RPC client that the init function returned.
port	[IN] logic port id;
clear	[IN] 1:reset port 0: not reset port
port_statistic/stat	[OUT] port statistic info ;

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Return
status_t.

adlink_port_statistic_clear

Prototype
status_t adlink_port_statistic_clear (CLIENT* clnt, UINT32 port);

Supported On

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Description
Clear port statistic info.

Parameters
clnt [IN] RPC client that the init function returned.
port [IN] logic port id;

Return
status_t.

adlink_port_status_set

Prototype
status_t adlink_port_status_set(CLIENT* clnt,adlink_port_status_set_cmd_t* cmd);

Supported On

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Description
Set port attributes.

Parameters
clnt [IN] RPC client that the init function returned.
cmd [IN] port attributes cmd;

Return
status_t.

adlink_nic_config_set ()

adlink_nic_config_get()

Prototype
status_t adlink_nic_config_set(CLIENT* clnt, adl_nic_config_set_cmd_t *cmd);

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

status_t adlink_nic_config_get(CLIENT* clnt, adl_nic_config_get_cmd_t *cmd,
adl_nic_config_info_t *info);

Supported On

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Description

The function ‘get’ is used to get configuration of a NIC, including running status and IP address. The function ‘set’ is used to configure a NIC.

Parameters

clnt [IN] RPC client that the init function returned.
cmd [IN] NIC name and operation code;
info [OUT] NIC information;

Return

status_t.

adlink_port_mirror_set ()

adlink_port_mirror_get()

Prototype

status_t adlink_port_mirror_set(CLIENT* clnt,int unit, UINT32 port, adlink_mirror_t *info);
status_t adlink_port_mirror_get(CLIENT* clnt, int unit,UINT32 port, adlink_mirror_t *info);

Supported On

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Description

The function ‘get’ is used to get mirror info of a port. The function ‘set’ is used to set the mirror info of a port.

Parameters

clnt [IN] RPC client that the init function returned.
Unit [IN] Fabric or Base unit.
info [IN/OUT] mirror information;

Return

status_t.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

adlink_port_status_notify_register ()

Prototype

status_t adlink_port_status_notify_register(CLIENT*rpc_clnt,int unit,
adlink_port_status_change_callback_fun func);

Supported On

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Description

The function is used to regist a call back function for port status. When the port status are changed, the function will be called automatically.

Parameters

clnt	[IN] RPC client that the init function returned.
Unit	[IN] Fabric or Base unit.
func	[IN] The call back function handle

Prototype

typedef void (*adlink_port_status_change_callback_fun)(CLIENT* clnt,int unit, int port, UINT32 event_mask, adlink_port_info_t *port_info);

Return

status_t.

adlink_port_id_get_by_name ()

adlink_port_name_get_by_id()

Prototype

status_t adlink_port_id_get_by_name (CLIENT* clnt,int unit,char* bcm_name,int* port_id);
status_t adlink_port_name_get_by_id (CLIENT* clnt,int unit,int port_id,char* bcm_name);

Supported On

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Description

The two functions are used for the conversion of bcm name and bcm port id.

Parameters

clnt	[IN] RPC client that the init function returned.
Unit	[IN] Fabric or Base unit.
Bcm_name	[IN/OUT] port bcm name, such as xe0;
Port_id	[IN/OUT] port id, such as 1;

Return

status_t.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.5 VLAN

This section describes the macro, enum, struct and functions for VLAN. They are defined in the header file “adlink_api_vlan.h”.

VLAN management interfaces include the following services:

- Add ports to and delete ports from a vlan
- Delete all ports of a vlan and destroy the vlan
- Set the egress tagging option(whether frames leave the switch VLAN tagged or untagged)

2.5.1 Macro Definitions

MAX_VLAN_MAC_CNT

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Maximum entry count of MAC based VLAN.

MAX_VLAN_NAME_SIZE

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Each VLAN can associate with a name. This is used to limit the name size.

2.5.2 Enum Type

adlink_vlan_tag_mode_t

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

```
typedef enum adlink_vlan_tag_mode_e{
    VLAN_MODE_UNTAGGED = 0,
    VLAN_MODE_TAGGED = 1,
} adlink_vlan_tag_mode_t;
```

Egress tagging options.

Fields	Description
VLAN_MODE_UNTAGGED	Untagged egress tagging mode
VLAN_MODE_TAGGED	Tagged egress tagging mode

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.5.3 Data Structure

adlink_vlan_info_t

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

```
typedef struct adlink_vlan_info_s {
    UINT8    enable;    /*admin status*/
    UINT8    rsv1[3]; /*for align purpose*/
    UINT64 pbmp;       /*port bitmap in this vlan*/
    UINT64 untag_pbmp;    /*port bitmap of untagged ports*/
    char     name[MAX_VLAN_NAME_SIZE];       /*vlan name. 31 characters at maxmium*/
}adlink_vlan_info_t;
```

Definitions of the VLAN table entry format.

Fields	Description
enable	Admin status. For cPCI-6S10 and MXN-0410, use API adlink_vlan_admin_status_set or VLAN enable command to make the VLAN enable. For other boards, after the VLAN created, it will at enable states automatically.
rsv1[3]	Alignment of data, useless.
pbmp	port bitmap in this vlan, include tag and untag ports(uint64 type at MXN-3610/MXN-4100)
untag_pbmp	port bitmap of untagged ports (uint64 type at MXN-3610/MXN-4100)
name[MAX_VLAN_NAME_SIZE]	VLAN name. 31 characters maximum

adlink_vlan_mac_t

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

```
typedef struct adlink_vlan_mac_s {
        UINT8    valid;
        UINT8    cos;
        UINT8    mac[MAC_SIZE];
        UINT16 vid;
        UINT8    rsv[2];
}adlink_vlan_mac_t;
```


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Definitions of the MAC based VLAN table entry format.

Fields	Description
valid	Vlan in using or not. 0—INVALID 1—VALID
cos	Priority value for the packets with the VLAN
mac[MAC_SIZE]	MAC address associate with the VLAN
vid	VLAN id, 1-4094

2.5.4 Functions

adlink_vlan_info_get ()

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Prototype

status_t adlink_vlan_info_get (CLIENT* clnt, int unit, int vlan_id, adlink_vlan_info_t *info);

Description

This function is used to get specific VLAN's information.

Parameters

clnt [IN] RPC client that the init function returned. Not support on cPCI-6S10 and MXN-0410

unit [IN] Fabric or Base chip unit;

vlan_id [IN] The VLAN ID on which to operate;

info [OUT] VLAN information;

Return

status_t

adlink_vlan_admin_status_set ()

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Prototype

status_t adlink_vlan_admin_status_set (int unit,int vlan, int status);

Description

This function is used to switch admin status of a VLAN. If the status is disabled, VLAN can be configured but not used. This function is used to created VLANs, set them to ENABLE.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Parameters

unit [IN] Fabric or Base chip unit;
vlan_id [IN] The VLAN ID on which to operate;
status [IN] Admin status. Use ‘ENABLE’ or ‘DISABLE’;

Return

status_t

adlink_vlan_destroy ()

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Prototype

status_t adlink_vlan_destroy (*CLIENT* clnt,*int unit,int vlan);

Description

This function is used to destroy an existing VLAN.

Parameters

clnt [IN] RPC client that the init function returned.
unit [IN] Fabric or Base chip unit;
vlan [IN] The VLAN ID to be destroyed;

Return

status_t.

adlink_vlan_member_add ()

adlink_vlan_member_del ()

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Prototype

status_t adlink_vlan_member_add (int unit, int vlan, UINT32 port_id, UINT32 mode);
status_t adlink_vlan_member_del (int unit, int vlan, UINT32 port_id);

Description

These two functions are used to add port into or remove port from a VLAN.

Parameters

unit [IN] Fabric or Base chip unit;
vlan [IN] The VLAN ID in which port member is configured.
port _id [IN] Port ID to be added/removed.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

mode [IN] Port is tagged or untagged in the VLAN. Use ‘VLAN_MODE_TAGGED’ or ‘VLAN_MODE_UNTAGGED’

Return
status_t.

adlink_vlan_member_add ()

adlink_vlan_member_del ()

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Prototype
status_t adlink_vlan_member_add(CLIENT* clnt,int unit,int vlan, UINT64 port_mask, UINT32 mode);
status_t adlink_vlan_member_del(CLIENT* clnt,int unit,int vlan, UINT64 port_mask);

Description
These two functions are used to add ports into or remove ports from a VLAN.

Parameters
unit [IN] Fabric or Base chip unit;
vlan [IN] The VLAN ID in which port member is configured.
port _mask [IN] Ports bitmap to be added/removed. Port mask is the same with port bitmap.
mode [IN] Port is tagged or untagged in the VLAN. Use ‘VLAN_MODE_TAGGED’ or ‘VLAN_MODE_UNTAGGED’

Return
status_t.

adlink_vlan_port_pvid_set ()

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Prototype
status_t adlink_vlan_port_pvid_set (int unit, int vlan, UINT32 port);

Description
This function is used to assign a port’s default VLAN tag for untagged packets.

Parameters
unit [IN] Fabric or Base chip unit;
vlan [IN] The default VLAN ID of a port;
port [IN] The port ID which will be assigned a default VLAN tag;

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Return
status_t

adlink_mac_based_vlan_add ()

adlink_mac_based_vlan_del ()

adlink_mac_based_vlan_get ()

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Prototype
status_t adlink_mac_based_vlan_add (CLIENT* clnt, int unit, int vlan, UINT8 *mac, UINT8 cos);
status_t adlink_mac_based_vlan_del (CLIENT* clnt, int unit, UINT8 *mac);
status_t adlink_mac_based_vlan_get (CLIENT* clnt, int unit, UINT8 *mac, UINT16 *vlan, UINT8 *cos);

Description
These functions are used to configure MAC base VLAN. MAC address can be associated with a VLAN id.Packet with specific MAC address with be located to a dedicate VLAN.

Parameters

Clnt	[IN] RPC client that the init function returned.Not support at cPCI-6S10 and MXN-0410
unit	[IN] Fabric or Base chip unit;
vlan	[IN] The VLAN ID which will be assigned to a specified MAC address;
mac	[IN] A pointer points to specified MAC address;
cos	[IN] Default COS value which will be filled in VLAN tag;

Return
status_t.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.6 Trunk

This section describes the macro, enum, struct and functions for trunk. They are defined in the header file “adlink_api_trunk.h”.

2.6.1 Macro Definitions

MIN_TRUNK_ID

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

Minimun trunk id

MAX_TRUNK_CNT

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

Maximum trunk count.

MAX_PORT_NUM_PER_TRUNK

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

Trunk group index range and member port count limit of a group.

MAX_TRUNK_PORT_NUM

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

Trunk group index range and member port count limit of a group.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.6.2 Enum Type

adlink_trunk_balance_type_t

prototype

```
typedef enum adlink_trunk_balance_type
{
    BALANCE_TYPE_DEFAULT    = 0,
    BALANCE_TYPE_SA         = 1,
    BALANCE_TYPE_DA         = 2,
    BALANCE_TYPE_SA_DA      = 3,
    BALANCE_TYPE_SIP        = 4,
    BALANCE_TYPE_DIP        = 5,
    BALANCE_TYPE_SIP_DIP    = 6,
    BALANCE_TYPE_ENHANCED= 7
}adlink_trunk_balance_type_t
```

Supported On

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

Define the codes of trunk port selection criteria. Packets will be dispatched to different ports based on the criteria.

PACKET_SELECT_FIELD_t

prototype

```
typedef enum PACKET_SELECT_FIELD_s
{
    SOURCE_MAC,
    DESTIONATION_MAC,
    ETHER_TYPE,
    VLAN_ID,
    VLAN_PRIORITY,
    SOURCE_IP,
    DESTIONATION_IP,
    PROTOCOL,
    TCP_UDP_SOURCE_PORT,
    TCP_UDP_DESTINATION_PORT,
    PACKET_SELECT_FIELD_END,
}PACKET_SELECT_FIELD_t;
```


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Supported On

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

Define the packet's fields which used to do port selection criteria. The default set of the fields is enabled, packets will be dispatched to different ports based on the criteria.

adlink_trunk_enhance_balance_type_t

Prototype

```
typedef enum adlink_trunk_enhance_balance_type_e
{
    ENHANCED_BALANCE_TYPE_INVALID   = 0,
    ENHANCED_BALANCE_TYPE_4TUPLE    = 1,
    ENHANCED_BALANCE_TYPE_5TUPLE    = 2,
}adlink_trunk_enhance_balance_type_t
```

Supported On

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

Define the trunk port selection criteria when balance type is selected as BALANCE_TYPE_ENHANCED. 4 Tuple means port is selected Source IP, Source Port, Destination IP, Destination Port. 5 Tuple means port is selected Source IP, Source Port, Destination IP, Destination Port, Protocol Type.

adlink_trunk_l4_port_flag_t

```
typedef enum adlink_trunk_l4_port_flag
{
    BALANCE_TYPE_L4_PORT_DISABLE    = 0,
    BALANCE_TYPE_L4_PORT_ENABLE     = 1,
}adlink_trunk_l4_port_flag_t;
```

Supported On

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Define the trunk port selection criteria when balance type is selected as BALANCE_TYPE_SIP or BALANCE_TYPE_DIP or BALANCE_TYPE_SIP_DIP. L4_Port_Disable means port is selected by "Source IP" or "Destination IP" or "Source IP and Destination IP", L4_Port_Enable means port is selected by "Source IP and Source Port" or "Destination IP and Destination Port" or "Source IP, Destination IP, Source Port, Destination Port".

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

adlink_trunk_mode_t

Prototype

```
typedef enum adlink_trunk_mode_e
{
    ADLINK_TRUNK_MODE_INVALID = 0,
    ADLINK_TRUNK_MODE_MANUAL  = 1,
    ADLINK_TRUNK_MODE_LACP    = 2,
}adlink_trunk_mode_t
```

Supported On

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

Define the work mode of the trunk. The manual trunk is a static one, the member ports of the trunk will not change. The LACP trunk is negotiated by LACP protocol, which ports will forward packets are decided by LACP protocol.

adlink_manual_trunk_port_status_t

Prototype

```
typedef enum adlink_manual_trunk_port_status_e
{
    ADLINK_MANUAL_TRUNK_PORT_DOWN   = 0,
    ADLINK_MANUAL_TRUNK_PORT_ACTIVE    = 1,
}adlink_ manual_trunk_port_status_t
```

Supported On

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

Define the member port status of the manual trunk. Down means the port link state is fault, active means the port link is up.

adlink_lacp_trunk_port_status_t

Prototype

```
typedef enum adlink_lacp_trunk_port_status_e
{
    ADLINK_LACP_TRUNK_PORT_DOWN   = 0,
    ADLINK_LACP_TRUNK_PORT_ACTIVE    = 1,
    ADLINK_LACP_TRUNK_PORT_STANDBY    = 2,
    ADLINK_LACP_TRUNK_PORT_NO_PARTNER = 3,
}adlink_lacp_trunk_port_status_t
```


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Supported On

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

Define the member port status of the LACP trunk. Down means the port link state is fault, active means the port link is up and the port is chosen to forward packets, standby means the port link is up but the port will not forward packets, no partner means the port can’t negotiate successfully with the partner port in the other device.

2.6.3 Data Structure

adlink_trunk_manual_port_t

Prototype

```
struct adlink_trunk_manual_port{
    unsigned int port_no;
    unsigned int port_state;
}
```

Supported On

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

Define the port member info of manual trunk.

adlink_trunk_manual_trunk_t

Protorype

```
struct adlink_trunk_manual_trunk{
    int unit;
    int tid;
    unsigned int port_num;
    adlink_trunk_manual_port_t port_info[MAX_TRUNK_PORT_NUM];
}
```

Supported On

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

typedef struct adlink_trunk_manual_trunk adlink_trunk_manual_trunk_t;
Define the trunk info of manual trunk.

adlink_trunk_lacp_port_t

Prototype

```
struct adlink_trunk_lacp_port{
    unsigned int port_no;
```


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
    unsigned int port_state;
    unsigned int port_priority;
    unsigned char lacp_state;
}
```

Supported On

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

Typedef struct adlink_trunk_lacp_port adlink_trunk_lacp_port_t;
Define the port member info of lacp trunk.

adlink_trunk_lacp_trunk_t

Prototype

```
struct adlink_trunk_lacp_trunk{
    int unit;
    int tid;
    unsigned int port_num;
    adlink_trunk_lacp_port_t port_info[MAX_TRUNK_PORT_NUM];
}
```

Supported On

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

typedef struct adlink_trunk_lacp_trunk adlink_trunk_lacp_trunk_t;
Define the trunk info of lacp trunks.

adlink_trunk_port_info_t

Prototype

```
struct adlink_trunk_port_info{
    unsigned int port_no;
    unsigned int port_status;
    unsigned int port_priority;
    unsigned char lacp_state;
}
```

Supported On

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

To describe the info of a port to be print out.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

adlink_trunk_info_t

Prototype

```
struct adlink_trunk_lacp_trunk{
    int unit;
    int tid;
    unsigned int port_num;
    adlink_trunk_port_info_t port_info[MAX_PORT_NUM_PER_TRUNK];
}
```

Supported On

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

To store main information of a lacp trunk.

adlink_trunk_balance_config_t

```
typedef struct adlink_trunk_balance_config_s
{
    int unit;
    PACKET_SELECT_FIELD_t packet_select_field;
    BOOL status;     //Enable or Disable
}adlink_trunk_balance_config_t;
```

Supported On

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

Trunk balance configuration of specific packet field.

adlink_trunk_balance_config_info_t

```
typedef struct adlink_trunk_balance_config_info_s
{
   adlink_trunk_balance_config_t trunk_hash_field_info [PACKET_SELECT_FIELD_END];

}adlink_trunk_balance_config_info_t;
```

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

Trunk balance info.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.6.4 Functions

adlink_trunk_ manual_trunk_add ()

Prototype

status_t adlink_trunk_manual_trunk_add (CLIENT* clnt,int unit, int trunk_id)

Supported On

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

Add a manual trunk.

Parameters

clnt [IN] RPC client that the init function returned.
unit [IN] Specified the switch chip,eg:Fabric/Base.
trunk_id [IN] Specified the target trunk, range <1-8>.

Return

status_t

adlink_trunk_lacp_trunk_add()

Prototype

status_t adlink_trunk_lacp_trunk_add(CLIENT* clnt,int unit, int Trunk_id)

Supported On

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

Add a lacp trunk.

Parameters

clnt [IN] RPC client that the init function returned.
unit [IN] Specified the switch chip,eg:Fabric/Base.
trunk_id [IN] Specified the target trunk ,range <1-8>.

Return

status_t

adlink_trunk_del()

Prototype

status_t adlink_trunk_del(CLIENT* clnt,int unit, int trunk_id)

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Supported On

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

remove a trunk.

Parameters

clnt [IN] RPC client that the init function returned.
unit [IN] Specified the switch chip,eg:Fabric/Base.
trunk_id [IN] Specified the target trunk ,range <1-8>.

Return

Status_t

adlink_trunk_lacp_priority_trunk_add ()

Prototype

status_t adlink_trunk_lacp_priority_trunk_add(CLIENT* clnt,int unit, int trunk_id, UINT32 sys_priority)

Supported On

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

add a lacp trunk with user-defined priority.

Parameters

clnt [IN] RPC client that the init function returned.
unit [IN] Specified the switch chip,eg:Fabric/Base.
trunk_id [IN] Specified the target trunk,range <1-8>.
sys_priority [IN] Specified the lacp trunk system priority value.

Return

status_t

adlink_trunk_manual_trunk_del ()

Prototype

status_t adlink_trunk_manual_trunk_del (CLIENT* clnt,int unit, int trunk_id)

Supported On

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

Remove a manual trunk.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Parameters

clnt [IN] RPC client that the init function returned.
unit [IN] Specified the switch chip,eg:Fabric/Base.
trunk_id [IN] Specified the target trunk,range <1-8>.

Return

status_t

adlink_trunk_lacp_port_del ()

Prototype

status_t adlink_trunk_lacp_trunk_del (CLIENT* clnt,int unit, int trunk_id);

Supported On

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

Remove a lacp trunk.

Parameters

clnt [IN] RPC client that the init function returned.
unit [IN] Fabric or Base chip unit.
trunk_id [IN] Trunk group id,range <1-8>.
sys_priority [IN] Priority of the trunk,0~65535,0 is highest.

Return

status_t

adlink_trunk_manual_port_add ()

Prototype

status_t adlink_trunk_manual_port_add (CLIENT* clnt,int unit, int trunk_id, UINT64 pbmp)

Supported On

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Prototype

status_t adlink_trunk_manual_port_add(CLIENT* clnt,int unit, int trunk_id, UINT32 port_id)

Supported On

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

Add a port to a manual trunk.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Parameters

clnt	[IN] RPC client that the init function returned.
unit	[IN] Fabric or Base chip unit.
trunk_id	[IN] Trunk group id,range <1-8>.
pbmp	[IN] a port bit map,specified the port set.
port_id	[IN] Specified a port ID.

Return

status_t

adlink_trunk_port_del()

Prototype

status_t adlink_trunk_port_del(CLIENT* clnt,int unit, int trunk_id, UINT64 pbmp);

Supported On

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

Remove a port from the trunk.

Parameters

clnt	[IN] RPC client that the init function returned.
unit	[IN] Fabric or Base chip unit.
trunk_id	[IN] Trunk group id,range <1-8>.
pbmp	[IN] a port bit map,specified the port set.

Return

status_t

adlink_trunk_manual_port_del ()

Prototype

status_t adlink_trunk_manual_port_del (CLIENT* clnt, int unit, int trunk_id, UINT32 port_id)

Supported On

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

Remove a port from a manual trunk.

Parameters

clnt	[IN] RPC client that the init function returned.
unit	[IN] Fabric or Base chip unit.
trunk_id	[IN] Trunk group id,range <1-8>.
port_id	[IN] Specified the port ID.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Return
status_t

adlink_trunk_lacp_port_add ()

Prototype
status_t adlink_trunk_lacp_port_add (CLIENT* clnt, int unit, int trunk_id, UINT64 pbmp)

Supported On

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Prototype
status_t adlink_trunk_lacp_port_add(CLIENT* clnt,int unit, int trunk_id, UINT32 port_id)

Supported On

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description
Add a port to a lacp trunk.

Parameters

clnt	[IN] RPC client that the init function returned.
unit	[IN] Fabric or Base chip unit.
trunk_id	[IN] Trunk group id,range <1-8>.
pbmp	[IN] A port bit map,specified the port set.
port_id	[IN] Specified the port ID.

Return
status_t

adlink_trunk_lacp_priority_port_add ()

Prototype
status_t adlink_trunk_lacp_priority_port_add (CLIENT* clnt,int unit, int trunk_id, UINT32 port_id, UINT32 port_priority)

Supported On

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description
Add a port to a lacp trunk with user-defined port priority.

Parameters

clnt	[IN] RPC client that the init function returned.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

unit	[IN] Fabric or Base chip unit;
trunk_id	[IN] Trunk group id; Start from ‘0’;
port_id	[IN] Port Id
port_priority	[IN] Priority of the trunk,0~65535,0 is highest

Return

status_t

adlink_trunk_lacp_pri_port_add ()

Prototype

status_t adlink_trunk_lacp_pri_port_add (CLIENT* clnt, int unit, int trunk_id, UINT64 pbmp, UINT32 pri)

Supported On

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

Add ports to a lacp trunk with user-defined port priority.

Parameters

clnt	[IN] RPC client that the init function returned.
unit	[IN] Fabric or Base chip unit.
trunk_id	[IN] Trunk group id,range <1-8>.
pbmp	[IN] A port bit map,specified the port set.
pri	[IN] Specified a lacp port priority.

Return

status_t

adlink_trunk_all_trunk_clear ()

Prototype

status_t adlink_all_trunk_clear (CLIENT* clnt, int unit)

Supported On

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

Remove all trunks.

Parameters

clnt	[IN] RPC client that the init function returned.
unit	[IN] Fabric or Base chip unit;

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Return
status_t

adlink_trunk_manual_trunk_get ()

Prototype
status_t adlink_trunk_manual_trunk_get(CLIENT* clnt,int unit, int trunk_id, adlink_trunk_manual_trunk_t* manual_trunk);

Supported On

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description
Get manual trunk information.

Parameters

clnt	[IN] RPC client that the init function returned.
unit	[IN] Fabric or Base chip unit;
trunk_id	[IN] Trunk group id,range <1-8>.
manual_trunk	[OUT] structure to store manual trunk information.

Return
status_t

adlink_trunk_lacp_trunk_get ()

Prototype
status_t adlink_trunk_lacp_trunk_get(CLIENT* clnt,int unit, int trunk_id, adlink_trunk_info_t* trunk_info)

Supported On

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

prototype
status_t adlink_trunk_lacp_trunk_get(CLIENT* clnt,int unit, int trunk_id, adlink_trunk_lacp_trunk_t* lacp_trunk);

Supported On

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description
This function is used to get trunk group information.

Parameters

clnt	[IN] RPC client that the init function returned.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

unit [IN] Fabric or Base chip unit;
trunk_id [IN] Trunk group id,range <1-8>.
lacp_trunk [OUT] structure to store lacp trunk information.
trunk_info [OUT] structure to store lacp trunk information.

Return
status_t

adlink_trunk_ balance_type _set ()

Prototype
status_t adlink_trunk_balance_type_set (CLIENT* clnt, int unit, int trunk_id, UINT32 balance_type)

Supported On

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description
Set the trunk load balance type.

Parameters
clnt [IN] RPC client that the init function returned.
unit [IN] Fabric or Base chip unit.
trunk_id [IN] Trunk group id,range <1-8>.
balance_type [IN] Specified a balance type, see enumerate in adlink_trunk_balance_type_t.

Return
status_t

adlink_trunk_ balance_type _get ()

Prototype
status_t adlink_trunk_balance_type_get(CLIENT* clnt,int unit, int trunk_id, UINT32 *balance_type)

Supported On

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description
Get the trunk load balance type.

Parameters
clnt [IN] RPC client that the init function returned.
unit [IN] Fabric or Base chip unit;

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

trunk_id [IN] Trunk group id; Start from ‘1’,invalid in MXN-3610/MXN-4100.
balance_type [OUT] Specified a balance type,see enumerate in adlink_trunk_balance_type_t.

Return
status_t.

adlink_trunk_enhanced_balance_mode_set ()

Prototype
status_t adlink_trunk_enhanced_balance_mode_set(CLIENT* clnt,int unit, UINT32 enhanced_mode)

Supported On

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description
These functions are used to control the enhanced port selection criteria, an ADLINK-defined hash method utilizing "mode7" in the Broadcom SDK (using adlink_trunk_balance_type_set to set balance type as BALANCE_TYPE_ENHANCED). Enhanced port selection criteria can be set to use 4-tuple (SrcIp, DstIp. SrcPort, DstPort) or 5-tuple (SrcIp, DstIp, SrcPort, DstPort, IpProtocol) to generate the hash value. Note that this is a system wide configuration and modification of this mode will affect all trunk groups that are using Enhanced port selection criteria.

Parameters
clnt [IN] RPC client that the init function returned.
unit [IN] Fabric or Base chip unit.
enhanced_mode [IN] 4-tuple/5-tuple.

Return
status_t

adlink_trunk_balance_config_set()
status_t adlink_trunk_balance_config_set(CLIENT *clnt,int unit, PACKET_SELECT_FIELD_t packet_select_field, BOOL set_config);

adlink_trunk_balance_config_get()
status_t adlink_trunk_balance_config_get(CLIENT *clnt,int unit, adlink_trunk_balance_config_info_t *trunk_balance_config_info);

adlink_trunk_enhanced_balance_mode_get()

Prototype
status_t adlink_trunk_enhanced_balance_mode_get(CLIENT* clnt,int unit, UINT32

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

*enhanced_mode)

Supported On

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

These functions are used to control the enhanced port selection criteria, an ADLINK-defined hash method utilizing "mode7" in the Broadcom SDK (using adlink_trunk_balance_type_set to set balance type as BALANCE_TYPE_ENHANCED). Enhanced port selection criteria can be set to use 4-tuple (SrcIp, DstIp. SrcPort, DstPort) or 5-tuple (SrcIp, DstIp, SrcPort, DstPort, IpProtocol) to generate the hash value. Note that this is a system wide configuration and modification of this mode will affect all trunk groups that are using Enhanced port selection criteria.

Parameters

clnt	[IN] RPC client that the init function returned.
unit	[IN] Fabric or Base chip unit.
enhanced_mode	[OUT] 4-tuple/5-tuple.

Return

status_t.

adlink_trunk_l4_port_enable()

Prototype

status_t adlink_trunk_l4_port_enable(CLIENT* clnt,int unit)

Supported On

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

These functions are used to control the port selection criteria detail, (using adlink_trunk_balance_type_set to set balance type as BALANCE_TYPE_SIP or BALANCE_TYPE_DIP or BALANCE_TYPE_SIP_DIP). L4 port disable means port is selected by “Source IP” or “Destination IP” or “Source IP and Destination IP”, L4 port enable means port is selected by “Source IP and Source Port” or “Destination IP and Destination Port” or “Source IP, Destination IP, Source Port, Destination Port”.

Parameters

clnt	[IN] RPC client that the init function returned.
unit	[IN] Fabric or Base chip unit;

Return

status_t

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

adlink_trunk_l4_port_disable()

Prototype

status_t adlink_trunk_l4_port_disable(CLIENT* clnt,int unit)

Supported On

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

These functions are used to control the port selection criteria detail, (using adlink_trunk_balance_type_set to set balance type as BALANCE_TYPE_SIP or BALANCE_TYPE_DIP or BALANCE_TYPE_SIP_DIP). L4 port disable means port is selected by “Source IP” or “Destination IP” or “Source IP and Destination IP”, L4 port enable means port is selected by “Source IP and Source Port” or “Destination IP and Destination Port” or “Source IP, Destination IP, Source Port, Destination Port”.

Parameters

clnt [IN] RPC client that the init function returned.
unit [IN] Fabric or Base chip unit;

Return

status_t

adlink_trunk_l4_port_get()

Prototype

status_t adlink_trunk_l4_port_get(CLIENT* clnt,int unit, UINT32 *flag)

Supported On

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

These functions are used to control the port selection criteria detail, (using adlink_trunk_balance_type_set to set balance type as BALANCE_TYPE_SIP or BALANCE_TYPE_DIP or BALANCE_TYPE_SIP_DIP). L4 port disable means port is selected by “Source IP” or “Destination IP” or “Source IP and Destination IP”, L4 port enable means port is selected by “Source IP and Source Port” or “Destination IP and Destination Port” or “Source IP, Destination IP, Source Port, Destination Port”.

Parameters

clnt [IN] RPC client that the init function returned.
unit [IN] Fabric or Base chip unit.
flag [IN|OUT] L4 enbale/disable.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Return
status_t

adlink_trunk_mode_get()

Prototype
adlink_trunk_mode_t adlink_trunk_mode_get(CLIENT* clnt,int unit, int trunk_id);

Supported On

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description
Get the trunk mode, manual or lacp.

Parameters

clnt	[IN] RPC client that the init function returned.
unit	[IN] Specified the switch chip,eg:Fabric/Base.
trunk_id	[IN] Specified the target trunk .

Return
adlink_trunk_mode_t

adlink_trunk_balance_config_set()

Prototype
adlink_trunk_balance_config_set(CLIENT *clnt,int unit, PACKET_SELECT_FIELD_t packet_select_field, BOOL set_config);

Supported On

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description
Set the packet field status which can be used as a hash key for trunk.

Parameters

clnt	[IN] RPC client that the init function returned.
unit	[IN] Specified the switch chip,eg:Fabric/Base.
packet_select_field	[IN] packet field
set_config	[IN] ENABLE/DISABLE

Return
status_t

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

adlink_trunk_balance_config_get()

adlink_trunk_balance_config_get(CLIENT *clnt,int unit, adlink_trunk_balance_config_info_t *trunk_balance_config_info);

Supported On

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

Get the packet field staus which can be used as a hash key for trunk.

Return

status_t

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.7 LBG

This section describes the macro, enum, struct and functions for LBG. They are defined in the header file “adlink_api_lbg.h”.

2.7.1 Macro Definitions

L34HASH_SIP

Supported On

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

Indicates hash field “source ip address” to operate on.

L34HASH_DIP

Supported On

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

Indicates hash field “destination ip address” to operate on.

L34HASH_L4SRC

Supported On

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

Indicates hash field “layer 4 source port” to operate on.

L34HASH_L4DST

Supported On

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

Indicates hash field “layer 4 destination port” to operate on.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

LBGACTIVEPORT

Supported On

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

Indicate lbg port set to active mode.

LBGBACKUPPORT

Supported On

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

Indicate lbg port set to backup mode..

L34HASH_PROT

Supported On

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

Indicates hash field “layer 4 protocol type” to operate on.

L34HASH_SYM

Supported On

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

Indicates hash field “layer 3, 4 symmetric” to operate on.

L34HASH_USETCP

Supported On

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

Indicates hash field “layer 4 protocol is TCP” to operate on.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

L34HASH_USEUDP

Supported On

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

Indicates hash field “layer 4 protocol is UDP” to operate on.

2.7.2 EnumType

None.

2.7.3 Data Structure

None.

2.7.4 Functions

adlink_lbg_add()

Prototype

status_t adlink_lbg_add(CLIENT* clnt, int unit, int lbgIndex, int numberOfBins)

Supported On

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

Create a load balance group.

Parameters

clnt	[IN] rpc client handler..
unit	[IN] switch chip number.
lbgIndex	[IN] LBG to be created.
numberOfBins	[IN] amount of hash bins, can be 0-16, 32, 64, 128, 256, 512, 1024, 2048, 4096. If 0, default value 128 will be used.

Return

status_t

ADLINK_E_NONE	if successful.
ADLINK_E_FAIL	if LBG cannot be created.
ADLINK_E_CONFIG	if LBG cannot be activated.
ADLINK_E_MALLOC	if memory cannot be allocated.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

adlink_lbg_del()

Prototype

status_t adlink_lbg_del(CLIENT *clnt, int unit, int lbgIndex)

Supported On

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

Delete a load balance group.

Parameters

clnt [IN] rpc client handler..
unit [IN] switch chip number.
lbgIndex [IN] LBG to be deleted.

Return

status_t
ADLINK_E_NONE if successful.
ADLINK_E_NOT_FOUND if LBG cannot be found.
ADLINK_E_FAIL if LBG cannot be deleted.

adlink_lbg_port_add()

Prototype

status_t adlink_lbg_port_add(CLIENT *clnt, int unit, int lbgIndex, int logicalPort, int portmode)

Supported On

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

Add a logical port to an LBG.

Parameters

clnt [IN] rpc client handler..
unit [IN] switch chip number.
lbgIndex [IN] target LBG index.
logicalPort [IN] logical port number to add.
Portmode [IN] logical port mode active or backup.

Return

status_t
ADLINK_E_NONE if successful.
ADLINK_E_NOT_FOUND if LBG cannot be found.
ADLINK_E_FAIL if port cannot be added.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

adlink_lbg_port_del()

Prototype

status_t adlink_lbg_port_del(CLIENT *clnt, int unit, int lbgIndex, int logicalPort)

Supported On

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

Delete a logical port from an LBG.

Parameters

clnt [IN] rpc client handler..
unit [IN] switch chip number.
lbgIndex [IN] target LBG index.
logicalPort [IN] logical port number to delete.

Return

status_t
ADLINK_E_NONE if successful.
ADLINK_E_NOT_FOUND if LBG cannot be found.
ADLINK_E_FAIL if port cannot be added.

adlink_lbg_hash34_set()

Prototype

status_t adlink_lbg_hash34_set(CLIENT *clnt, int unit, int l34HashField, int l34HashFieldVal)

Supported On

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

Set L34 hashing properties for all LBGs.

Parameters

clnt [IN] rpc client handler..
unit [IN] switch chip number.
l34HashField [IN] hash source field, please refer to macros "L34HASH_*" in header file "adlink_api_lbg.h".
l34HashFieldVal [IN] bool value for l34HashField, 1-on, 0-off.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Return

status_t

ADLINK_E_NONE	if successful.
ADLINK_E_INTERNAL	if current hash config cannot be obtained.
ADLINK_E_FAIL	if port cannot be added.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.8 Layer-2

This section describes the macro, enum, struct and functions for Layer-2. They are defined in the header file “adlink_api_l2.h”.

2.8.1 Macro Definitions

MAC_SIZE

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

MAC addresses data size in bytes.

ADLINK_L2_xxx

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

L2 entry flag masks. Flags are special attributes of the MAC entry. For details, please refer to the definitions in “adlink_api_l2. h”.

2.8.2 Enum Type

None.

2.8.3 Data Structure

adlink_mac_entry_t

Prototype

```
typedef struct adlink_mac_entry_s
{
    UINT32   port;              /*port ID associated with the entry, start from 1*/
    UINT16   vid;               /*vlan id*/
    UINT8    rsv0[2];           /*reserved*/
    UINT32   flag;              /*entry flag, ADLINK_L2_XXX*/
    UINT8    mac[MAC_SIZE];     /*mac address*/
    UINT8    rsv1[2];           /*reserved*/
} adlink_mac_entry_t;
```


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

Define the entry format of the MAC table.

proxy_mac_item_t

Prototype

```
typedef struct proxy_mac_item_s{
    UINT32 flag;            /*ADLINK_L2_xxx*/
    UINT32 port;            /*port id associated to the mac address*/
    UINT16 vid;             /*VLAN id associated to the mac address*/
    UINT8  mac[MAC_SIZE];/*mac address*/
    }proxy_mac_item_t;
```

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

Define the struct of mac item.

proxy_mac_dump_t

Prototype

```
typedef struct proxy_mac_dump_s{
    UINT8  end;                                  /*indicate whether retrive finished*/
    UINT8  cnt;                                  /*retrived item number in the list*/
    UINT16 last_index;                           /*last item index in mac table*/
    proxy_mac_item_t item[MAX_MAC_LIST_SIZE];    /*retrived mac items*/
    }proxy_mac_dump_t;
```

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

Define the struct for l2 mac table dumping.

2.8.4 Functions

adlink_mac_entry_add ()

Prototype

```
status_t adlink_mac_entry_add (CLIENT* clnt, adlink_mac_entry_t *entry);
```


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

These functions are used to change the MAC address table. Add/delete an item or clear all items.

Parameters

clnt [IN] RPC client that the init function returned.

entry [IN] Entry that will be added or deleted;

Return

status_t.

adlink_mac_entry_del ()

Prototype

status_t adlink_mac_entry_del (CLIENT* clnt, adlink_mac_entry_t *entry);

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

These functions are used to change the MAC address table. Add/delete an item or clear all items.

Parameters

clnt [IN] RPC client that the init function returned.

entry [IN] Entry that will be added or deleted.

Return

status_t.

adlink_mac_table_clear ()

Prototype

status_t adlink_mac_table_clear (CLIENT* clnt);

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

These functions are used to change the MAC address table. Add/delete an item or clear all items.

Parameters

clnt [IN] RPC client that the init function returned.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

entry [IN] Entry that will be added or deleted;

Return
status_t.

adlink_mac_table_dump ()

Prototype
status_t adlink_mac_table_dump (CLIENT* clnt, proxy_index_t *start_index,
proxy_mac_dump_t *list);

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description
This function is used to dump the entire MAC address table.

Parameters
clnt [IN] RPC client that the init function returned.
start_index [IN] Start index of MAC address table;
list [OUT] Retrieved MAC entries;

Return
status_t

adlink_mac_filter_add ()

adlink_mac_filter_del ()

Prototype
status_t adlink_mac_filter_add(CLIENT* clnt,adlink_mac_entry_t *entry);
status_t adlink_mac_filter_del(CLIENT* clnt,adlink_mac_entry_t *entry);;

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description
This function is used to add/delete the filter mac.

Parameters
clnt [IN] RPC client that the init function returned.
entry [IN] mac entry for mac filter;

Return
status_t

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.9 Access Control List

This section describes the macro, enum, struct and functions for Access Control List (ACL). They are defined in the header file “adlink_api_acl.h”.

2.9.1 Macro Definitions

MAX_QUALIFIER_NUM

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

The ACL supports multiple qualifiers in a rule. This macro defines the maximum qualifier number allowed in a rule.

MAX_UDF_DATA_CNT

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

MAX count of UDF data can be added.

RULE_INFO_ACL_CONFIG

RULE_INFO_QOS_CONFIG

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Define the rule is ACL or QOS. Now only ACL can be used.

2.9.2 Enum Type

adlink_qualifier_type_t

```
typedef enum adlink_qualifier_type
{
    INPORT = 1,      //port id,data32, only support one port, it is waste for
MXN-3610/MXN-4100.
    INPORTS,//data64, port bits map
    SRCIP,   //data32
    DSTIP,         //data32
    IPPROTOCOL, //data8
```


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
        DSCP, //data8
        L4SRCPORT,   //data16
        L4DSTPORT,   //data16
        SRCMAC,      //data48
        DSTMAC,      //data48
        //SRCTRUNK,    //data16
        ETHERTYPE,   //data16
        INNERVLAN,   //data16
        OUTERVLAN,   //data16
        UDF,     //data16,
        STAGELOOKUP, //No data, customer can't use
    }adlink_qualifier_type_t;
```

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Qualifier Type	Description
INPORT	Filter for the port ID a packet comes from
INPORTS	Filter for the port ID a packet comes from
SRCIP	Filter for source IP address of a packet
DSTIP	Filter for destination IP address of a packet
IPPROTOCOL	Filter for IP protocol (such as 0x06 for TCP) of a packet
L4SRCPORT	Filter for L4 source port of a packet
L4DSTPORT	Filter for L4 destination port of a packet
SRCMAC	Filter for source MAC address of a packet
DSTMAC	Filter for destination MAC address of a packet
SRCTRUNK	Filter for the trunk ID a packet comes from
ETHERTYPE	Filter for Ethernet type (such as 0x0806 for ARP) of a packet
INNERVLAN	Filter for the inner VLAN ID of a packet
OUTERVLAN	Filter for the outer VLAN ID of a packet
UDF	Filter for the user defined field
STAGELOOKUP	Internal retention

Table 2. Return Code Description

adlink_action_type_t

```
    typedef enum adlink_action_type
    {
        REDIRECTPORT = 1*64+1,     //port id,data32
        REDIRECTTRUNK,   //trunk id,data16
        DROP,                //no param
        OUTERVLANNEW,     //vlan id,data16
        REMARK_USR_PRIORITY, //Internal retention
```


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
        MIRROR_INGRESS,        //Internal retention
        MIRROR_EGRESS,         //Internal retention
        REMARK_DSCP,           //Internal retention
        REMARK_TOS,            //Internal retention
        SET_DISCARD_ABILITY,   //Internal retention
        ASSIGN_OUTPUT_QUEUE, //Internal retention
    } adlink_action_type_t;
```

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Action Type	Description
REDIRECTPORT	Redirect the packet to a port (cpu0, xe0, xe1, xe2…).
REDIRECTTRUNK	Redirect the packet to a trunk
DROP	Drop the packet

Table 3. Action Type Description

adlink_udf_offset_type_t

```
typedef enum adlink_udf_offset_type
{
    UDF_PACKETSTART = 1,
    UDF_OUTERL3HEAD = 2,
    UDF_INNERL3HEAD = 3,
    UDF_OUTERL4HEAD = 4,
    UDF_INNERL4HEAD = 5
}adlink_udf_offset_type_t;
```

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Offset type	Description
UDF_PACKETSTART	Offset from the packet head.
UDF_OUTERL3HEAD	Offset from the packet outer l3 head.
UDF_INNERL3HEAD	Offset from the packet inner l3 head
UDF_OUTERL4HEAD	Offset from the packet outer l4 head
UDF_INNERL4HEAD	Offset from the packet inner l4 head

Table 4 Action Type Description

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

adlink_priority_type_t

```
typedef enum adlink_priority_type
{
    HIGH_PRIORITY = 0,
    LOW_PRIORITY = 1,
}adlink_priority_type_t;
```

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

2.9.3 Data Structure

adlink_udf_data_info_t

```
typedef struct adlink_udf_data_info
{
    int         index;
    UINT32      offset_base;
    UINT16      offset;
    UINT16      ethtype;
    UINT8       ipproto;
    UINT8       resv;
}adlink_udf_data_info_t;
```

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Udf data info struct.

adlink_acl_data_t

```
typedef struct adlink_acl_data
{
    UINT8       data128[16];//data128 use, such as ipv6 addr
    UINT64      data64;       //data64 use,such as inports pbmp
    UINT8       data48[6];    //data48 use, such as mac addr
    UINT32      data32;       //data32 use, such as ip addr
    UINT16      data16;       //data16 use, such as l4port
    UINT8       data8;        //data8 use, such as ip protocol
} adlink_acl_data_t;
```


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

For different qualifiers and actions, there are different data lengths. For example, an IP address is 32 bits long, a MAC address is 48 bits long, and an L4 port ID is 16 bits long. This data structure was defined in order to accommodate these different situations. Careful attention must be paid to refer to the appropriate member according to the actual situation. The appropriate members for existing qualifiers and actions are listed in the tables above.

adlink_acl_qualifier_info_t

```
typedef struct adlink_acl_qualifier_info
{
    adlink_qualifier_type_t     type;
    UINT16                      udf_id;//only used for type=UDF
    adlink_acl_data_t           data;//use the correct member for different types
    adlink_acl_data_t           mask;//use the correct member for different types
} adlink_acl_qualifier_info_t;
```

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Qualifier Type	Data	mask
INPORT	data32(port id)	None
INPORTS	Data64(port bit maps,cpu0 is bit 0,xe0 is bit 1,…)	
SRCIP	data32	data32
DSTIP	data32	data32
IPPROTOCOL	Data8	Data8
L4SRCPORT	Data16	Data16
L4DSTPORT	Data16	Data16
SRCMAC	Data48	Data48
DSTMAC	Data48	Data48
SRCTRUNK	Data16	Data16
ETHERTYPE	Data16	Data16
INNERVLAN	Data16	Data16
OUTERVLAN	Data16	Data16
UDF	Data16	Data16

Table 5. Data Descriptions for Different Qualifier Types

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

adlink_acl_action_info_t

```
typedef struct adlink_acl_action_info
{
    adlink_action_type_t    type;
    adlink_acl_data_t       param0;//union,use the correct member for different types
    adlink_acl_data_t       param1;//union,use the correct member for different types
} adlink_acl_action_info_t;
```

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Action Type	Param0	Param1
REDIRECTPORT	data32 (port id)	None
REDIRECTTRUNK	Data16 (trunk id)	None
DROP	None	None

Table 6. Data Descriptions for Different Action Types

adlink_acl_rule_info_t

```
typedef struct adlink_acl_rule_info
{
    UINT16 priority;        // the priority of the ACL rule
    UINT16 qualifier_num;   //the number of qualifier of the ACL rule
    adlink_acl_qualifier_info_t qualifier[MAX_QUALIFIER_NUM];
                                //record the qualifier info
    adlink_acl_action_info_t action;// record the action info
} adlink_acl_rule_info_t;
```

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

2.9.4 Functions

adlink_acl_rule_add ()

Prototype

status_t adlink_acl_rule_add (CLIENT* clnt, adlink_acl_rule_info_t *rule_info,UINT16 *rule_id);

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Description
This function is used to add an ACL rule.

Parameters
clnt [IN] RPC client that the init function returned.
rule_info [IN] The ACL rule info needed to be added.
rule_id [OUT] The pointer for saving the rule ID.

Return
status_t.

adlink_acl_rule_delete ()

Prototype
status_t adlink_acl_rule_delete (CLIENT* clnt, UINT16 rule_id);

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Description
This function is used to delete an ACL rule.

Parameters
clnt [IN] RPC client that the init function returned.
rule _id [IN] The ID of the ACL rule that want to delete.

Return
status_t.

adlink_acl_rule_clear ()

Prototype
status_t adlink_acl_clear (CLIENT* clnt);

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Description
This function is used to clear the ACL table in the system. After executing this function, all ACL rules in the system will be lost.

Parameters
clnt [IN] RPC client that the init function returned.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Return
status_t

adlink_acl_rule_get ()

Prototype
status_t adlink_acl_rule_get (CLIENT* clnt, UINT16 rule_id, adlink_acl_rule_info_t *rule_info);

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Description
This function is used to get the detail information of an ACL rule.

Parameters
clnt [IN] RPC client that the init function returned.
rule_id [IN] The ID of the rule that want to get information.
rule_info [OUT] The pointer for saving the rule information.

Return
status_t

adlink_udf_data_set ()

adlink_udf_data_delete()

adlink_udf_data_get()

Prototype
status_t adlink_udf_data_set(CLIENT* clnt,adlink_udf_data_info_t *udf_info);
status_t adlink_udf_data_delete(CLIENT* clnt,UINT16 index);
status_t adlink_udf_data_get(CLIENT* clnt,UINT16 index,adlink_udf_data_info_t *udf_info);

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Description
This functions are used for udf.

Parameters
clnt [IN] RPC client that the init function returned.
Index [IN] udf date index.
udf_info [IN/OUT] The udf data information.

Return
status_t

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.10 Spanning Tree Protocol

This section describes the macro, enum, struct and functions for Spanning Tree Protocol (STP). They are defined in the header file “adlink_api_stp.h”. This feature related APIs are only applied to cPCI-6S10 and MXN-0410. STP feature not support trunk port now.

Spanning Tree Protocol management interfaces include the following services:

- Create and delete bridge.
- Add ports to and delete ports from a bridge.
- Set bridge status, priority, forward delay time, hello time, max age time.
- Set port path cost, port priority
- Get bridge info
- Print bridge and port STP info

2.10.1 Macro Definitions

MAX_BR_NUM

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

This is used to define the maxiumum bridge numbers in our system. Actually our system now just support only one bridge.

MAX_BR_PORTS

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

This defines the maxiumum ports that can be add to a bridge.

2.10.2 Enum Type

stp_status

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

```
enum stp_status{
      STP_ON = 1,
      STP_OFF = 2,
};
```


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

The stp status for bridge.

Fields	Description
STP_ON	Bridge status is on
STP_OFF	Bridge status is off

2.10.3 Data Structure

bridge_id_t

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

```
typedef struct bridge_id_s{
        unsigned char prio[2];
        unsigned char addr[6];
 }bridge_id_t;
```

Define the bridge id info.

Fields	Description
prio	Birdge priority
addr	MAC address for the bridge.

stp_para_t

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

```
union stp_para_u{
        int stp_status;
        int br_pri;
        int path_cost;
        int port_pri;
        uint time;
        };
```

Defines the stp parameters to use.

Fields	Description
stp_status	STP staus.
br_pri	Bridge priority.
path_cost	Ports path cost.
port_pri	Ports priority.
time	Time value for hello time, forward dealy or max age.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

proxy_stp_para_item_t

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

```
typedef struct proxy_stp_para_item_s{
        char    bridge_name[32];
        int       port_id;
        union stp_para_u stp_para;
}proxy_stp_para_item_t;
```

Defines the STP parameters for a port or bridge

Fields	Description
bridge_name	Bridge name
port_id	Port id
stp_para	See union stp_para_u

adlink_br_info_t

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

```
typedef struct adlink_br_info_s{
    char                br_name[32];
    bridge_id_t         br_id;
    unsigned char    stp_enabled;
    int port_num;
    struct port_name port_name_list[MAX_BR_PORTS];
}adlink_br_info_t;
```

Defines the bridge info.

Fields	Description
br_name	Bridge name
br_id	Birdge id
stp_enabled	Bridge status
port_num	Num of ports in the bridge
port_name_list	Port name list

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

adlink_br_list_info_t

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

```
typedef struct adlink_br_list_info_s{
    int br_num;
    adlink_br_info_t br_info[MAX_BR_NUM];
}adlink_br_list_info_t;
```

Defines the bridge list info of the whole system.

Fields	Description
br_num	Num of bridges
br_info	Bridge related info

bridge_stp_info

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

```
struct bridge_stp_info
{
        bridge_id_t designated_root;
        bridge_id_t bridge_id;
        unsigned root_path_cost;
        struct time_val max_age;
        struct time_val hello_time;
        struct time_val forward_delay;
        struct time_val bridge_max_age;
        struct time_val bridge_hello_time;
        struct time_val bridge_forward_delay;
        unsigned short    root_port;
        unsigned char stp_enabled;
        unsigned char topology_change;
        unsigned char topology_change_detected;
        struct time_val ageing_time;
        struct time_val hello_timer_value;
        struct time_val tcn_timer_value;
        struct time_val topology_change_timer_value;
        struct time_val gc_timer_value;
};
```


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Defines detailed STP info of bridge.

Fields	Description
designated_root	Designated root bridge id
bridge_id	Bridge id
root_path_cost	Root path cost
max_age	For internal use
hello_time	BPDU hello time
forward_delay	Forward delay time
bridge_max_age	Bridge max age time
bridge_hello_time	Bridge hello time
bridge_forward_delay	Birdge forward delay time
root_port	Root port
stp_enabled	STP status
topology_change	Topology change flag.
topology_change_detected	Topology change detect flag.
ageing_time	For internal use
hello_timer_value	Timer to send BPDU hello packets
tcn_timer_value	TCN timer for internal use
topology_change_timer_value	Topology change timer for internal use
gc_timer_value	GC timer value for internal use

br_stp_port_info

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

```
struct br_stp_port_info
 {
        unsigned port_no;
        bridge_id_t designated_root;
        bridge_id_t designated_bridge;
        unsigned short port_id;
        unsigned short designated_port;
        unsigned char priority;
        unsigned char top_change_ack;
        unsigned char config_pending;
        unsigned char state;
        unsigned path_cost;
        unsigned designated_cost;
        struct time_val message_age_timer_value;
        struct time_val forward_delay_timer_value;
        struct time_val hold_timer_value;
 };
```


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Defines the detailed port info of bridge.

Fields	Description
port_no	Port index of the current port for the bridge. Start from 1
designated_root	Designated root bridge id
designated_bridge	Designated bridge id
port_id	Port id for the bridge, It consist of port no and priority.
designated_port	Designed port id
priority	Port priority
top_change_ack	For internal use
config_pending	For internal use
state	Port stp state. Disable/listening/learning//forwarding/ blocking
path_cost	Ports' path cost
designated_cost	Designated path cost
message_age_timer_value	For internal use
forward_delay_timer_value	Forward delay timer
hold_timer_value	Hold timer for internal use

adlink_stp_modify_t

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

```
typedef struct adlink_stp_modify_s {
    int             unit;
    char            bridge_name[32];
    int             portbmp;

}adlink_stp_modify_t;
```

Used for STP modify operation.

Fields	Description
unit	Fabric or Base unit
bridge_name	Specific the bridge to operate.
portbmp	Specific the ports to operate

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

adlink_stp_para_set_t

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

```
typedef struct adlink_stp_para_set_s{
        int       unit;
        char    bridge_name[32];
        int       port_id;
        union    stp_para_u stp_para;
}adlink_stp_para_set_t;
```

Used to set STP parameters.

Fields	Description
unit	Fabric or Base unit
bridge_name	Specific the bridge to operate.
port_id	Specific the port to operate
stp_para	STP related parameters

adlink_br_stp_info_t

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

```
typedef struct adlink_br_stp_info_s{
      int bridge_find;
      struct bridge_stp_info bridge_stp;
      int port_num;
      struct port_name br_port_name[MAX_BR_PORTS];
      struct br_stp_port_info port_info[MAX_BR_PORTS];
}adlink_br_stp_info_t;
```

Defines the bridge ports infos.

Fields	Description
bridge_find	Bridge exist or not.
bridge_stp	Bridge info
port_num	Port in the bridge
Br_port_name	Port name
Port_info	Port infos

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.10.4 Functions

adlink_stp_bridge_add ()

adlink_stp_bridge_del ()

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Prototype

status_t adlink_stp_bridge_add(adlink_stp_modify_t *status)
status_t adlink_stp_bridge_del(adlink_stp_modify_t *status)

Description

This function is used to add/delete the specific bridge.

Parameters

status [IN] Modification info include unit and bridge name.

Return

status_t.

adlink_stp_itf_add ()

adlink_stp_itf_del ()

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Prototype

status_t adlink_stp_itf_add (adlink_stp_modify_t *status)
status_t adlink_stp_itf_del (adlink_stp_modify_t *status)

Description

This function is used to add/delete the ports from the specific bridge.

Parameters

status [IN] Modification info include unit ,bridge name and ports.

Return

status_t.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

adlink_stp_status_set ()

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Prototype

status_t adlink_stp_status_set (adlink_stp_para_set_t *status)

Description

This function is used to set STP status.

Parameters

status [IN]. Parameters include unit, bridge name and stp status

Return

status_t.

adlink_stp_br_pri_set ()

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Prototype

status_t adlink_stp_br_pri_set (adlink_stp_para_set_t *status)

Description

This function is used to set bridge priority.

Parameters

status [IN]. Parameters include unit, bridge name and bridge priority.
The bridge priority shoud in range <0-61440> and in step of 4096

Return

status_t.

adlink_stp_fd_time_set()

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Prototype

status_t adlink_stp_fd_time_set (adlink_stp_para_set_t *status)

Description

This function is used to set bridge priority.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Parameters

status [IN]. Parameters include unit, bridge name and forward delay time.
The forward delay time should in range <2-30> and follow
(2*(Hello-time +1) <= Max-Age <= 2*(Forward-Delay - 1))

Return

status_t.

adlink_stp_hello_time_set ()

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Prototype

status_t adlink_stp_hello_time_set (adlink_stp_para_set_t *status)

Description

This function is used to set hello time.

Parameters

status [IN]. Parameters include unit, bridge name and hello time.
The hello time should in range <1-10> and follow
(2*(Hello-time +1) <= Max-Age <= 2*(Forward-Delay - 1))

Return

status_t.

adlink_stp_max_age_time_set ()

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Prototype

status_t adlink_stp_fd_time_set (adlink_stp_para_set_t *status)

Description

This function is used to set max age time.

Parameters

status [IN]. Parameters include unit, bridge name and max age time.
The max age time should in range <6-40> and follow
(2*(Hello-time +1) <= Max-Age <= 2*(Forward-Delay - 1))

Return

status_t.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

adlink_stp_path_cost_set ()

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Prototype

status_t adlink_stp_path_cost_set (adlink_stp_para_set_t *status)

Description

This function is used to set port path cost.

Parameters

status [IN]. Parameters include unit, bridge name, port id and path cost. adlink_port_id_get_by_bcmname () is used to get port id from port name. The path cost should in range <1-100>

Return

status_t.

adlink_stp_port_pri_set ()

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Prototype

status_t adlink_stp_port_pri_set (adlink_stp_para_set_t *status)

Description

This function is used to set port priority.

Parameters

status [IN]. Parameters include unit, bridge name, port id and port priority. The path cost should in range <1-100>

Return

status_t.

adlink_stp_br_list_info_get ()

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Prototype

status_t adlink_stp_br_list_info_get (adlink_br_list_info_t *stat)

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Description

This function is used to get bridge list information. The info contains the bridge name, brige state and bridge ports. Refer to “adlink_br_list_info_t” for more details.

Parameters

status [IN]. Pointer to the bridge list infos.

Return

status_t.

adlink_stp_br_list_info_print ()

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Prototype

status_t adlink_stp_br_list_info_print (adlink_br_list_info_t *stat)

Description

This function is used to print bridge list informations.

Parameters

status [IN]. Pointer to the bridge list infos.

Return

status_t.

adlink_br_stp_info_get ()

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Prototype

status_t adlink_br_stp_info_get (char *br_name, adlink_br_stp_info_t *stat)

Description

This function is used to get bridge detail STP informations by bridge name.

Parameters

br_name [IN]. bridge name

status [OUT]. Pointer to the bridge STP information.

Return

status_t.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

adlink_br_info_print ()

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Prototype

status_t adlink_br_info_print (char *br_name, adlink_br_stp_info_t *stat)

Description

This function is used to print bridge's detail STP informations (Not include port's info).

Parameters

br_name [IN]. bridge name

status [IN]. Pointer to the bridge STP information.

Return

status_t.

adlink_br_port_info_print ()

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Prototype

status_t adlink_br_port_info_print (adlink_br_stp_info_t *stat)

Description

This function is used to print port's detail STP informations.

Parameters

status [IN]. Pointer to the port's STP information.

Return

status_t.

adlink_br_stp_info_print ()

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Prototype

status_t adlink_br_stp_info_print (char *br_name, adlink_br_stp_info_t *stat)

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Description

This function is used to print current bridge's detail STP informations. It is equalivalent to using adlink_br_info_print () and adlink_br_port_info_print ().

Parameters

status [IN]. Pointer to the bridge's detail STP information, including bridge info and its port info.

Return

status_t.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.11 Rapid Spanning Tree Protocol

This section describes the macro, enum, struct and functions for Rapid Spanning Tree Protocol (RSTP). They are defined in the header file “adlink_api_stp_rstp.h”. RSTP feature not support trunk/LAG port now.

This feature related API include the following services:

- Enable and disable the RSTP feature.
- Force this feature work at STP or RSTP mode.
- Set bridge priority, forward delay time, hello time, max age time.
- Set port path cost, port priority, port STP status, port P2P status, port edge status, port mcheck function.
- Get current spanning tree global info and port info.

2.11.1 Macro Definitions

STP_PORT_NAME_MAX_LEN

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

This is used to define the max length of the STP/RSTP port name.

2.11.2 Enum Type

INFO_IS_T

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

```
typedef enum
{
    Mine,
    Aged,
    Received,
    Disabled
} INFO_IS_T;
```

The origin/state of the port’s spanning tree information (port info) held for ports.

Fields	Description
Mine	The port has been derived from the root port for the bridge (with the addition of root port cost information). This includes the possibility that the root port is “Port 0”, i.e., the bridge is the root bridge for the bridged Local Area Network (LAN).

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Fields	Description
Aged	The root bridge has been aged out.
Received	The port has received current (not aged out) information from the designated bridge for the attached LAN (a point-to-point bridge link being a special case of a LAN).
Disabled	Port is disabled.

RCVD_MSG_T

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

```
typedef enum
{
      SuperiorDesignateMsg = ADLINK_FALSE,
      RepeatedDesignateMsg = ADLINK_TRUE,
      ConfirmedRootMsg,
    OtherMsg
} RCVD_MSG_T;
```

This is used to set the result of the rcvBPDU procedure. Mostly used for internal state machine.

Fields	Description
SuperiorDesignateMsg	Superior designate message.
RepeatedDesignateMsg	Repeated designate message
ConfirmedRootMsg	Confirmed root message
OtherMsg	Other message

This is used to set the result of the rcvBPDU procedure.

PORT_ROLE_T

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

```
typedef enum
{
      DisabledPort = 0,
      AlternatePort,
      BackupPort,
      RootPort,
      DesignatedPort,
      MasterPort,
      NonStpPort
} PORT_ROLE_T;
```

This is used to assign Port Role.

Fields	Description
DisabledPort	Disabled Port
AlternatePort	Alternate Port
BackupPort	Backup Port
RootPort	Root Port

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Fields	Description
DesignatedPort	Designated Port
MasterPort	Master Port
NonStpPort	Non-STP Port

ADMIN_P2P_T

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

```
typedef enum
{
    P2P_FORCE_TRUE,
    P2P_FORCE_FALSE,
    P2P_AUTO,
} ADMIN_P2P_T;
```

This is used to set port p2p mode

Fields	Description
P2P_FORCE_TRUE	Force port work at P2P mode
P2P_FORCE_FALSE	Force port work not at P2P mode
P2P_AUTO	Force port work at P2P auto mode

UID_STP_MODE_T

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

```
typedef enum
{
    STP_DISABLED ,
    STP_ENABLED,
} UID_STP_MODE_T;
```

Enums the feature status.

Fields	Description
STP_DISABLED	Feature disabled
STP_ENABLED	Feature enabled
STP_REENABLE	Feature Reenable.

PROTOCOL_VERSION_T

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

```
typedef enum PROTOCOL_VERSION
{
    FORCE_STP_COMPAT = 0,
    NORMAL_RSTP = 2,
    FORCE_MSTP = 3
```


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

} PROTOCOL_VERSION_T;

This is used to indicate the force protocol version parameter for the bridge.

Fields	Description
FORCE_STP_COMPAT	STP mode
NORMAL_RSTP	RSTP mode(Default mode)
FORCE_MSTP	MSTP mdoe(Not support now)

INTF_TYPES_t

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

```
typedef enum INTF_TYPES_s
{
      PHYSICAL_INTF = 3,
      LAG_INTF,
      VLAN_INTF,
      CPU_INTF,
} INTF_TYPES_t;
```

The interface type.

Fields	Description
PHYSICAL_INTF	Physical type interface
LAG INTF	Lag type interface
VLAN INTF	Vlan type interface
CPU INTF	CPU type interface

2.11.3 Data Structure

BRIDGE_ID

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

```
typedef struct bridge_id_def
 {
      unsigned short        prio;
      unsigned char         addr[6];
 } BRIDGE_ID;
```

Define the bridge id info.

Fields	Description
prio	Birdge priority
addr	MAC address for the bridge.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

TIMEVALUES_T

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

```
typedef struct timevalues_t
{
    unsigned short MessageAge;
    unsigned short MaxAge;
    unsigned short ForwardDelay;
    unsigned short HelloTime;
    char    Remaining_Hops ;
} TIMEVALUES_T;
```

Define the spanning tree time values.

Fields	Description
MessageAge	Birdge priority
MaxAge	MAC address for the bridge.
ForwardDelay	Forward delay time
HelloTime	Hello time
Remaining_Hops	Not used

PRIO_VECTOR_T

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

```
typedef struct prio_vector_t
{
    BRIDGE_ID root_bridge;
    unsigned long root_path_cost;
    BRIDGE_ID region_root_bridge ;
    unsigned long region_root_path_cost;
    BRIDGE_ID design_bridge;
    PORT_ID     design_port;
    PORT_ID     bridge_port;
} PRIO_VECTOR_T;
```

The port priority vector is the spanning tree priority vector held for the port when the reception of BPDUs and any pending update of information have been completed.

Fields	Description
root_bridge	Root Bridge Identifier, the Bridge Identifier of the Bridge believed to be the Root by the transmitter

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Fields	Description
root_path_cost	Root Path Cost, to that Root Bridge from the transmitting Bridge
region_root_bridge	Not used
region_root_path_cost	Not used
design_bridge	Bridge Identifier, of the transmitting Bridge
design_port	Port Identifier, of the Port through which the message was transmitted
bridge_port	Port Identifier, of the Port through which the message was received (where relevant)

usp

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

```
typedef struct usp_s
{
    UINT32 unit;
    UINT32 port;
} usp;
```

unit and port info structure.

Fields	Description
unit	Switch unit
port	For MXN-3610/MXN-4100, this is the port id. For other blades, the port id is a logical index for the port. For Fabric unit, port = port id – MIN_FABRIC_PORT_ID; For Base unit, port= port id – MIN_BASE_PORT_ID. (This make port value start from 0)

usl

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

```
typedef struct usl_s
{
    UINT32 unit;
    UINT32 lag_id;
} usl;
```


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Unit and lag id info structure.

Fields	Description
unit	Switch unit
lag_id	Trunk id.

intf_info_t

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

```
typedef struct intf_info_s
{
    INTF_TYPES_t intf_type;
    union
    {
        usl usl_info;
        usp usp_info;
    } port_info;
} intf_info_t;
```

Interface info structure.

Fields	Description
Intf_type	Switch unit
port_info	Physical port info or lag interface info

adlink_rstp_port_info_t

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

```
typedef struct adlink_rstp_port_info
{
    UINT16          port_index;
    intf_info_t     intf_info;
    char            port_name[STP_PORT_NAME_MAX_LEN + 1];
    PORT_ID         port_id;
    INFO_IS_T       infoIs;
    RCVD_MSG_T      rcvdMsg;
    PORT_ROLE_T     role;
    PORT_ROLE_T     selectedRole;
    ADMIN_P2P_T     adminPointToPointMac;
    PRIO_VECTOR_T   designPrio;
    TIMEVALUES_T    designTimes;
```


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
    PRIO_VECTOR_T       portPrio;
    TIMEVALUES_T        portTimes;
    RCVD_MSG_T          rcvdInfo ;
    unsigned long       adminPCost;
    unsigned long       operPCost;
    unsigned long       operSpeed;
    unsigned long       usedSpeed;
    int                 LinkDelay;
    unsigned long       rx_cfg_bpdu_cnt;
    unsigned long       rx_rstp_bpdu_cnt;
    unsigned long       rx_tcn_bpdu_cnt;
    unsigned long       uptime;
    unsigned long     forwarding_uptime;
} adlink_rstp_port_info_t;
```

Define the port RSTP related info.

Fields	Description
port_index	Port id
Intf_info	Port or lag info (include unit info)
port_name	Port's name
port_id	The port identifier. It forms with port priority and port id. See the IEEE802.1D RSTP part for more details.
infoIs	A variable that takes the values Mine, Aged, Received, or Disabled, to indicate the origin/state of the Port's Spanning Tree information (portInfo) held for the Port. See the IEEE802.1D RSTP part for more details.
rcvdMsg	An internal bool value used for communicate with state machines. See the IEEE802.1D RSTP part for more details.
role	The Rapid Spanning Tree Algorithm assigns one of the following Port Roles to each Bridge Port: Root Port, Designated Port, Alternate Port, Backup Port, or Disabled Port. See the IEEE802.1D RSTP part for more details.
selectedRole	The newly computed role for the port. See the IEEE802.1D RSTP part for more details.
adminPointToPointMac	P2P mode, TRUE/FALSE/AUTO
admin_non_stp	0 --- STP port 1 -- Non-STP port
adminEnable	Port link up or not
portEnabled	A boolean. Set if the Bridge's MAC Relay Entity and Spanning Tree Protocol Entity can use the

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Fields	Description
	MAC Service provided by the Port's MAC entity to transmit and receive frames to and from the attached LAN, i.e., portEnabled is TRUE if and only if: a) MAC_Operational is TRUE; and b) Administrative Bridge Port State for the Port is Enabled; and c) AuthControlledPortStatus is Authorized [if the port is a network access port (IEEE Std 802.1X)]. See the IEEE802.1D RSTP part for more details.
adminEdge	Present in implementations that support the identification of edge ports. See the IEEE802.1D RSTP part for more details.
operEdge	A boolean. The value of the operEdgePort parameter, as determined by the operation of the Bridge Detection state machine. See the IEEE802.1D RSTP part for more details.
agree	Set if syncedis set for all other Ports. An RST BPDU with the Agreement flag set is transmitted and proposedis reset when agreeis first set, and when proposed is set. See the IEEE802.1D RSTP part for more details.
agreed	Set when an RST BPDU is received with a Port Role of Root, Alternate, or Backup Port, the Agreement flag set, and a message priority the same or worse than the port priority.When agreedis set, the Designated Port knows that its neighbouring Bridge has confirmed that it can proceed to the Forwarding state without further delay. See the IEEE802.1D RSTP part for more details.
forward	State machine forward state. See the IEEE802.1D RSTP part for more details.
forwarding	Port forwarding state. See the IEEE802.1D RSTP part for more details.
initPm	Internal used
learn	State machine learn state. See the IEEE802.1D RSTP part for more details.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Fields	Description
learning	State machine learing state. See the IEEE802.1D RSTP part for more details.
mcheck	May be set by management to force the Port Protocol Migration state machine to transmit RST BPDUs for a MigrateTime period, to test whether all STP Bridges on the attached LAN have been removed and the Port can continue to transmit RSTP BPDUs. See the IEEE802.1D RSTP part for more details.
newInfo	Set if a BPDU is to be transmitted. Reset by the Port Transmit state machine. See the IEEE802.1D RSTP part for more details.
proposed	Set when an RST BPDU with a Designated Port role and the Proposal flag set is received. If agreeis not set, proposedcauses syncto be set for all other Ports.of the Bridge. See the IEEE802.1D RSTP part for more details.
proposing	Set by a Designated Port that is not Forwarding, and conveyed to the Root Port or Alternate Port of a neighboring Bridge in the Proposal flag of an RST BPDU.See the IEEE802.1D RSTP part for more details.
rcvdBpdu	Set by system dependent processes, this variable notifies the Port Receive state machine When a valid Configuration, TCN, or RST BPDU is received on the Port. Reset by the Port Receive state machine. See the IEEE802.1D RSTP part for more details.
rcvdRSTP	RSTP BPDU received. See the IEEE802.1D RSTP part for more details.
rcvdSTP	STP BPDU received. See the IEEE802.1D RSTP part for more details.
rcvdTc	Received Topology change. See the IEEE802.1D RSTP part for more details.
rcvdTcAck	Received Topology change Ack. See the IEEE802.1D RSTP part for more details.
rcvdTcn	Received TCN. See the IEEE802.1D RSTP part for more details.
reRoot	Root port role state machine reRoot state. See the IEEE802.1D RSTP part for more details.
reselect	Port role selection state machine at reselect state. See the IEEE802.1D RSTP part for more

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Fields	Description
	details.
selected	Port role selection state machine at selected state. See the IEEE802.1D RSTP part for more details.
sendRSTP	Port Protocol Migration state machine's sendRSTP state. See the IEEE802.1D RSTP part for more details.
sync	If set the Port transitions to Discarding, unless it is an Edge Port or syncedis set.
synced	Set when the Port is Discarding or agreedis set.
tc	Port Timers state machine 's tick parameter
tcAck	Set if a Configuration Message with a topology change acknowledge flag set is to be transmitted
tcProp	A boolean. Set by the Topology Change state machine of any other Port, to indicate that a topology change should be propagated through this Port.
updtInfo	A boolean. Set by the Port Role Selection state machine to tell the Port Information state machine that it should copy designated Priority to port Priority and designatedTimes to portTimes.
wasInitBpdu	Port received BPDU packets.
p2p_recompute	P2P enter recomputed state
operPointToPointMac	P2P state
designPrio	The first four components of the Port's designated priority vector value. The fifth component of the designated priority vector value is portId. See the IEEE802.1D RSTP part for more details.
designTimes	The designatedTimes variable comprises the set of timer parameter values (Message Age, Max Age, Forward Delay, and Hello Time) that used to update Port Times when updtInfo is set. See the IEEE802.1D RSTP part for more details.
portPrio	The first four components of the Port's port priority vector value.
portTimes	The portTimes variable comprises the Port's timer parameter values (Message Age, Max Age, Forward Delay, and Hello Time). These

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Fields	Description
	timer values are used in BPDUs transmitted from the Port.
rcvdInfo	Set to the result of the rcvInfo() procedure. See the IEEE802.1D RSTP part for more details.
adminPCost	The method to set port path cost. AUTO or FORCE.
operPCost	Port's operate path cost.
operSpeed	Port's operate speed.
usedSpeed	Port's speed.
LinkDelay	A link delay timer for port.
rx_cfg_bpdu_cnt	Received config BPDU number
rx_rstp_bpdu_cnt	Received rstp BPDU number
rx_tcn_bpdu_cnt	Received tcn bpdu number
uptime	Feature enabled time
forwarding_uptime	Forwarding up time

adlink_rstp_br_port_t

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

```
typedef struct adlink_rstp_br_port_s
{
    int br_port_id;
    intf_info_t port_info;
} adlink_rstp_br_port_t;
```

Define the RSTP bridge port info

Fields	Description
br_port_id	The port id that related to the bridge. This is automatically assigned when port add to bridge.
port_info	Port or lag info (include type and unit info)

adlink_rstp_br_unit_port_t

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

```
typedef struct adlink_rstp_br_unit_port_s
{
    int unit;
```


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
    int br_port_id;
} adlink_rstp_br_unit_port_t;
```

Define the RSTP unit and bridge port id info.

Fields	Description
unit	Switch unit
br_port_id	The port id that related to the bridge. This is automatically assigned when port add to bridge

adlink_rstp_br_info_t

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

```
typedef struct adlink_rstp_br_info_t
{
    UID_STP_MODE_T          admin_state;    //STP bridge admin state
    PROTOCOL_VERSION_T      ForceVersion; //Force Version
    BRIDGE_ID                   BrId; //Bridge id
    TIMEVALUES_T            BrTimes; //Bridge timers
    PORT_ID                     rootPortId;//Root port id
    PRIO_VECTOR_T           rootPrio; //Root priority
    TIMEVALUES_T            rootTimes; //Root times
    unsigned long           timeSince_Topo_Change; //time since topology change
    unsigned long           Topo_Change_Count;//time change count
    unsigned char           Topo_Change; //topo change flag
    int port_nums;          //Port numbers belongs to bridge
    adlink_rstp_br_port_t   port_info[32];      //Port info
} adlink_rstp_br_info_t;
```

Define RSTP bridge info.

Fields	Description
admin_state	STP bridge admin state
ForceVersion	Feature work mode, STP or RSTP, not support MSTP
BrId	Bridge id
BrTimes	Bridge timers
rootPortId	Root port id
rootPrio	Root priority
rootTimes	Root timer values
timeSince_Topo_Change	Time since topology change
Topo_Change_Count	Topology change count
Topo_Change	Topology change

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Fields	Description
port_nums	Number of ports
port_info	Port info

2.11.4 Functions

adlink_stp_br_status_set ()

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Prototype

status_t adlink_stp_br_status_set(CLIENT* clnt, UINT32 unit, UINT32 br_config_status);

Description

This function is used to enable/disable the RSTP feature. After this feature eabled, all of the port will be added to the spanning-tree in default.

Parameters

clnt [IN] RPC client that the init function returned.
unit [IN] switch unit
br_config_status [IN] bridge config status, ENABLE or DISABLE

Return

status_t.

adlink_stp_br_force_version_set ()

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Prototype

status_t adlink_stp_br_force_version_set(CLIENT* clnt, UINT32 unit, UINT32 br_force_version);

Description

This function is used to set the force version of the bridge.

Parameters

clnt [IN] RPC client that the init function returned.
unit [IN] switch unit
br_force_version [IN] FORCE_STP_COMPAT or NORMAL_RSTP

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Return
status_t.

adlink_stp_br_priority_set ()

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Prototype
status_t adlink_stp_br_priority_set(CLIENT* clnt, UINT32 unit, UINT32 br_priority);

Description
This function is used to set RSTP bridge priority;

Parameters

clnt	[IN] RPC client that the init function returned.
unit	[IN]switch unit
br_priority	[IN] bridge priority

Return
status_t.

adlink_stp_br_hello_time_set ()

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Prototype
status_t adlink_stp_br_hello_time_set (CLIENT* clnt, UINT32 unit, UINT32 br_hello_time);

Description
This function is used to set bridge hello time.

Parameters

clnt	[IN] RPC client that the init function returned.
unit	[IN] switch unit
br_hello_time	[IN] bridge hello time

Return
status_t.

adlink_stp_br_max_age_set ()

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Prototype
status_t adlink_stp_br_max_age_set (CLIENT* clnt, UINT32 unit, UINT32 br_max_age);

Description
This function is used to set maximum age timer.

Parameters

clnt	[IN] RPC client that the init function returned.
unit	[IN] switch unit
br_max_age	[IN]bridge max age time

Return
status_t.

adlink_stp_br_forward_delay_set ()

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Prototype
status_t adlink_stp_br_forward_delay_set (CLIENT* clnt, UINT32 unit, UINT32 br_fwd_delay);

Description
This function is used to set forward delay timer.

Parameters

clnt	[IN] RPC client that the init function returned.
unit	[IN] Switch unit
br_fwd_delay	[IN] forward delay timer

Return
status_t.

adlink_stp_port_path_cost_set ()

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Prototype
status_t adlink_stp_port_path_cost_set (CLIENT* clnt, UINT32 unit, UINT32 port_id, UINT32 path_cost);

Description
This function is used to set port path cost.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Parameters

clnt	[IN] RPC client that the init function returned.
unit	[IN] unit
port_id	[IN] port id
path_cost	[IN] path cost

Return

status_t.

adlink_stp_port_priority_set ()

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Prototype

status_t adlink_stp_port_priority_set (CLIENT* clnt, UINT32 unit, UINT32 port_id, UINT32 priority);

Description

This function is used to set port priority.

Parameters

clnt	[IN] RPC client that the init function returned.
unit	[IN] switch unit
port id	[IN] port id
priority	[IN]Port priority

Return

status_t

adlink_stp_port_status_set ()

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Prototype

status_t adlink_stp_port_status_set(CLIENT* clnt, UINT32 unit, UINT32 pbmp, UINT32 port_stp_status);

Description

The function is used to set the port's STP status.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Parameters

clnt	[IN] RPC client that the init function returned.
unit	[IN] switch unit
pbmp	[IN] input port bitmap
port_stp_status	[IN] enable:2 disable:1

Return

status_t.

adlink_stp_port_p2p_set()

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Prototype

status_t adlink_stp_port_p2p_set(CLIENT* clnt, UINT32 unit, UINT32 port_id, UINT32 port_p2p_status)

Description

This function is used to set the port's p2p mode.

Parameters

clnt	[IN] RPC client that the init function returned.
unit	[IN] switch unit
port id	[IN] port id
port_p2p_status	[IN] port P2P status 1:yes 2:no 3:auto

Return

status_t.

adlink_stp_port_edge_set ()

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Prototype

status_t adlink_stp_port_edge_set (CLIENT* clnt, UINT32 unit, UINT32 port_id, UINT32 port_edge_status);

Description

The function is used to set port edge status.

Parameters

clnt	[IN] RPC client that the init function returned.
unit	[IN] switch unit

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

port id	[IN] port id
port_edge_status	[IN]yes:1 no:0.

Return

status_t.

adlink_stp_port_mcheck_set ()

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Prototype

status_t adlink_stp_port_mcheck_set (CLIENT* clnt, UINT32 unit, UINT32 port_id)

Description

The function is used to set port mcheck.

Parameters

clnt	[IN] RPC client that the init function returned.
unit	[IN] switch unit
port id	[IN] port id

Return

status_t

adlink_stp_info_get ()

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Prototype

status_t adlink_stp_info_get(CLIENT* clnt, UINT32 unit, adlink_rstp_br_info_t *rstp_br_info);

Description

The function is used to set port mcheck.

Parameters

clnt	[IN] RPC client that the init function returned.
unit	[IN] switch unit
rstp_br_info	[OUT] RSTP bridge info

Return

status_t.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

adlink_stp_br_port_info_get ()

Supported on

aTCA-3710	aTCA-N700	aTCA-3430	MXN-3610/MXN-4100	cPCI-6S10	MXN-0410

Prototype

status_t adlink_stp_br_port_info_get (CLIENT*clnt, int unit, int br_port_id,
adlink_rstp_port_info_t *rstp_port_info);

Description

The function is used to set port mcheck.

Parameters

clnt	[IN] RPC client that the init function returned.
unit	[IN] switch unit
int	[IN]bridge port id
rstp_br_info	[OUT] RSTP port info

Return

status_t.

This section describes the macro, enum, struct and functions for Rapid Spanning Tree Protocol (RSTP). They are defined in the header file “adlink_api_stp_rstp.h”.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.12 LLDP

This section describes the macro,enum,data structure functions for Link Layer Discovery Protocol.They are defined in the head file “adlink_api_lldp.h”

2.12.1 Macro Definitions

LLDP_VIA_LEN

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

The max length of the via string of neighbor infomation

LLDP_CHISSIS_SUBTYPE_LEN

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

The max length of the string used to identify the chassis type

LLDP_IFNAME_LEN

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

The max length of the name string of interfaces

LLDP_AGE_LEN

Support on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

The max length of the age string of neighbor infomation

LLDP_SYSNAME_LEN

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

The max length of system name string

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

LLDP_MGMTIP_LEN

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

The max length of the management IP address string

LLDP_MAC_ID_LEN

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

The max length of the ID string identified with MAC address

LLDP_CHASSISDESC_LEN

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

The max length of the string describing the system chassis

LLDP_SYSDESC_LEN

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

The max length of the string used to identify the system description

2.12.2 Enum Type

OPERATION_TYPE_t

Prototype

```
typedef enum OPERATION_TYPE
{
    DEFAULET_NUM = 0,
    LLDP_PAUSE,
    LLDP_RESUME,
    LLDP_UPDATE,
    LLDP_TX_INTERVAL,
    LLDP_TX_HOLD,
}OPERATION_TYPE_t;
```


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

To enumerate operation types of lldp cli commands.

2.12.3 Data Structure

adlink_lldp_neighbor_t

Prototype

```
typedef struct adlink_lldp_neighbor_s{
    char    itfacename[LLDP_IFNAME_LEN];
    char    via[LLDP_VIA_LEN];
    UINT16 rid;
    char    age[LLDP_AGE_LEN];
    char    chassisidsubtype[LLDP_CHISSIS_SUBTYPE_LEN];
    char    chassisid[LLDP_MAC_ID_LEN];
    char    chassisdesc[LLDP_CHASSISDESC_LEN];
    char    sysname[LLDP_SYSNAME_LEN];
    char    mgmtip[LLDP_MGMTIP_LEN];
    UINT8   portsubtype;
    char    portsubid[LLDP_MAC_ID_LEN];
    char    portdesc[LLDP_PORTDESC_LEN];
}adlink_lldp_neighbor_t;
```

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

Structure used to transmit neighbor information by cli command “show neighbor”

adlink_lldp_configure_t

Prototype

```
typedef struct adlink_lldp_configure_s{
    UINT32 lldpMessageTxInterval;
    UINT32 lldpMessageTxHoldMultiplier;
    UINT32 lldpReinitDelay;
    UINT32 lldpTxDelay;
    UINT32 lldpNotificationInterval;
}adlink_lldp_configure_t;
```


LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

Structure for lldp base configuration information.

adlink_lldp_stats_rem_table_t

Prototype

```
typedef struct adlink_lldp_stats_rem_table_s{
    UINT32 lldpStatsRemTablesLastChangeTime;
    UINT32 lldpStatsRemTablesDeletes;
    UINT32 lldpStatsRemTablesDrops;
    UINT32 lldpStatsRemTablesAgeout;
}adlink_lldp_stats_rem_table_t;
```

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

Structure for lldp statistics of remote tables.

adlink_lldp_local_system_t

Prototype

```
typedef struct adlink_lldp_local_system_s{
    UINT32 lldpLocChassisIdSubtype;
    UINT8 lldpLocChassisId[MAC_SIZE];
    char lldpLocSysName[48];
    char lldpLocSysDesc[LLDP_SYSDESC_LEN];
    UINT8 lldpLocSysCapSupported;
    UINT8 lldpLocSysCapEnabled;
}adlink_lldp_local_system_t;
```

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

Structure for lldp local system data.

adlink_lldp_stats_tx_port_t

Prototype

```
typedef struct adlink_lldp_stats_tx_port_s {
    UINT32 lldpStatsTxPortNum;
    char   lldpStatsTxPortName[MAX_PORT_NUM][IFNAMSIZ];
```


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
    UINT32 lldpStatsTxPortFramesTotal[MAX_PORT_NUM];
}adlink_lldp_stats_tx_port_t;
```

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

Structure for lldp tx port stats.

stats_rx_port_content_t;

Prototype

```
typedef struct stats_rx_port_content {
    UINT32 lldpStatsRxPortFramesDiscardedTotal;
    UINT32 lldpStatsRxPortFramesErrors;
    UINT32 lldpStatsRxPortFramesTotal;
    UINT32 lldpStatsRxPortTLVsDiscardedTotal;
    UINT32 lldpStatsRxPortTLVsUnrecognizedTotal;
    UINT32 lldpStatsRxPortAgeoutsTotal;
}stats_rx_port_content_t;
```

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

Structure for one lldp rx port stats.

adlink_lldp_stats_rx_port_t

Prototype

```
typedef struct adlink_lldp_stats_rx_port_s {
    UINT32 lldpStatsRxPortNum;
    stats_rx_port_content_t stats_rx_port_content_st[MAX_PORT_NUM];
}adlink_lldp_stats_rx_port_t;
```

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

Structure forall lldp rx port stats.

adlink_lldp_local_port_t

Prototype

```
typedef struct adlink_lldp_local_port_s {
    UINT16 lldpLocPortIdSubtype;
```


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
    UINT8     lldpLocPortId[MAC_SIZE];
    char      lldpLocPortDesc[LLDP_PORTDESC_LEN];
}adlink_lldp_local_port_t;
```

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

Structure about local ports component.

adlink_lldp_rem_port_t

Prototype

```
typedef struct adlink_lldp_rem_port_s {
    UINT16 lldpRemChassisIdSubtype;
    UINT16 lldpRemPortIdSubtype;
    UINT8 lldpRemChassisId[MAC_SIZE];
    UINT8 lldpRemPortId[MAC_SIZE];
    char lldpRemPortDesc[LLDP_PORTDESC_LEN];
    char lldpRemSysName[48];
    char lldpRemSysDesc[LLDP_SYSDESC_LEN];
    UINT8 lldpRemSysCapSupported;
    UINT8 lldpRemSysCapEnabled;
}adlink_lldp_rem_port_t;
```

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

Suructure about information of a remote port component.

adlink_lldp_rem_man_addr_t

Prototype

```
typedef struct adlink_lldp_rem_man_addr_s{
    UINT16 lldpRemManAddrSubtype;
    UINT16 lldpRemManAddrIfId;
    UINT32 lldpRemManAddrOID;
}adlink_lldp_rem_man_addr_t;
```

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

Structure for management address information about a chassis component.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

operation_set_t

Prototype

```
typedef struct operation_set
{
    OPERATION_TYPE_t type;
    UINT32       val;
}operation_set_t;
```

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

Structure for lldp set operation by cli command.

2.12.4 Functions

adlink_lldp_neibor_info_get_by_port_id()

Prototype

status_t adlink_lldp_neighbor_info_get_by_port_id(CLIENT* clnt,UINT32 unit,UINT32 port, adlink_lldp_neighbor_t *info);

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

Function used to get neighbor infomation by port id.

Parameters

clnt [IN] RPC client that the init function returned.
unit [IN] switch unit
port [IN] logic port id
info [IN] lldp neighbor info

Return

status_t

adlink_lldp_configuration_get()

Prototype

status_t adlink_lldp_configuration_get(CLIENT* clnt,UINT32 unit,adlink_lldp_configure_t *info);

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Description
Function used to get lldp base configuration info.

Parameters
clnt [IN] RPC client that the init function returne.
unit [IN] switch unit
info [IN] lldp base configration info

Return
status_t

adlink_lldp_stats_rem_tables_get()

Prototype
status_t adlink_lldp_stats_rem_tables_get(CLIENT* clnt,UINT32 unit,
adlink_lldp_stats_rem_table_t *info);

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description
Function used to get lldp statistics info about remote tables

Parameters
clnt [IN] RPC client that the init function returned
unit [IN] switch unit
info [IN] lldp statistics info about remote table

Return
status_t

adlink_lldp_local_system_get()

Prototype
status_t adlink_lldp_local_system_get(CLIENT* clnt,UINT32 unit,adlink_lldp_local_system_t
*info);

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description
Function used to get local system info

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Parameters

clnt [IN] RPC client that the init function returned
unit [IN] switch unit
info [IN] lldp local system info

Return

status_t

adlink_lldp_stats_tx_port_get()

Prototype

status_t adlink_lldp_stats_tx_port_get(CLIENT* clnt,UINT32 unit,adlink_lldp_stats_tx_port_t *info);

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

Function used to get lldp tx port statistics

Parameters

clnt [IN] RPC client that the init function returned
unit [IN] switch unit
info [IN] lldp statistics info about tx ports

Return

status_t

adlink_lldp_stats_rx_port_get()

Prototype

status_t adlink_lldp_stats_rx_port_get(CLIENT* clnt,UINT32 unit,adlink_lldp_stats_rx_port_t *info);

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

Function used to get lldp rx port statistics

Parameters

clnt [IN] RPC client that the init function returned
unit [IN] switch unit
info [IN] lldp statistics info about rx ports

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Return
status_t

status_t adlink_lldp_local_port_get()

Prototype
status_t adlink_lldp_local_port_get(CLIENT* clnt,UINT32 unit,UINT32 port,
adlink_lldp_local_port_t *info);

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description
Function used to get lldp local port table info

Parameters
clnt [IN] RPC client that the init function returned
unit [IN] switch unit
port [IN] logic port id
info [IN] lldp local port table info

Return
status_t

adlink_lldp_rem_chassis_port_get()

Prototype
status_t adlink_lldp_rem_chassis_port_get(CLIENT* clnt,UINT32 unit,UINT32 port,
adlink_lldp_rem_port_t *info);

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description
Function used to get lldp remote ports and chassis info

Parameters
clnt [IN] RPC client that the init function returned
unit [IN] switch unit
port [IN] logic port id
info [IN] lldp remote ports info

Return
status_t

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

adlink_lldp_rem_man_addr_get()

Prototype

status_t adlink_lldp_rem_man_addr_get(CLIENT* clnt,UINT32 unit,UINT32 port,
adlink_lldp_rem_man_addr_t *info);

Description

Function used to get lldp management address info of a remote chassis component

Parameters

clnt [IN] RPC client that the init function returned
unit [IN] switch unit
port [IN] logic port id
info [IN] lldp remote management info

Return

status_t

adlink_lldp_operation_set()

Prototype

status_t adlink_lldp_operation_set(CLIENT* clnt,operation_set_t operation,int *result);

Description

Function used to do the lldp set operation

Parameters

clnt [IN] RPC client that the init function returned
operation [IN]Operation type and value
result [IN] To store the operation excu result

Return

status_t

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.13 VRRP

This section describes the macro,enum,data structure functions for Link Layer Discovery Protocol.They are defined in the head file “adlink_api_vrrp.h”

2.13.1 Macro Definitions

MIN_VRID

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

The minimum virtual router ID.

MAX_VRID

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

The max virtual router ID.

VRRP_MAX_IP_NUM

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

The max virtual ip address number one virtual router can been added to.

2.13.2 EnumType

vrrp_setType

Prototype

```
typedef enum {
    PRI_SET,
    PREEMPTION_SET,
    AUTHMODE_SET,
}vrrp_setType;
```

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Description

To enumerate vrrp setting operation type.

vrrp_state_enum

Prototype

```
typedef enum {
    VRRP_MASTER = 1,
    VRRP_BACKUP = 2,
}vrrp_state_enum
```

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

To enumerate virtual router state.

2.13.3 Data Structure

vrrp_setOpt_u

Prototype

```
typedef union {
    int vrid;
    vrrp_state_enum state;
    int priority;
    BOOL preemption;
    UINT32 ipaddr;
    int    auth_mode;
    char macaddr[ETHER_ADDR_LENGTH];
}vrrp_setOpt_u
```

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

The union for setting operation to transfer the value.

vrrp_vRoute_info_t

Prototype

```
typedef struct {
    int vlanid;
    int    vrid;
    int    priority;
```


Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
    int   naddr;
    int   adver_int;
    int   state;
    UINT32 ipaddr[VRRP_MAX_IP_NUM];
    BOOL preempt;
    char vmac[ETHER_ADDR_LENGTH];
    int result; //0-success;-1 no vrid
} vrrp_vRoute_info_t
```

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

To transfer the virtual router information.

2.13.4 Functions

adlink_vrrp_vRoute_add()

Prototype

status_t adlink_vrrp_vRoute_add(CLIENT* clnt, int unit, int vlanid, int vrid,UINT32 vip)

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

To add a new virtual router on a vlan-interface.

Parameters

clnt	[IN] RPC client that the init function returned.
unit	[IN] Fabric or Base chip unit.
vlanid	[IN]VLAN id.
vrid	[IN]virtual router id set by user.
vip	[IN]virtual ip address.

Return

status_t

adlink_vrrp_vRoute_rmv()

Prototype

status_t adlink_vrrp_vRoute_rmv(CLIENT* clnt, int unit, int vrid)

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

To remove a virtual router on a vlan-interface.

Parameters

clnt [IN] RPC client that the init function returned.
unit [IN] Fabric or Base chip unit.
vrid [IN] virtual router id set by user.

Return

status_t

adlink_vrrp_config_set()

Prototype

status_t adlink_vrrp_config_set(CLIENT* clnt, int unit, int vrid, vrrp_setType opt_type, vrrp_setOpt_u opt_value)

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

To do vrrp attribute setting.

Parameters

clnt [IN] RPC client that the init function returned.
unit [IN] Fabric or Base chip unit.
vrid [IN] virtual router id set by user.
opt_type [IN]indicate a setting operation type, PRI_SET, PREEMPTION_SET,or others to expand
opt_value [IN]a value crosspond with the opt_type.

Return

status_t

adlink_vrrp_vRoute_get()

Prototype

status_t adlink_vrrp_vRoute_get(CLIENT* clnt, int unit, int vrid,vrrp_vRoute_info_t *vrt_info)

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Description

To get the current vrrp arttribute and state information.

Parameters

clnt	[IN] RPC client that the init function returned.
unit	[IN] Fabric or Base chip unit.
vrid	[IN] virtual router id set by user.
vrt_info	[IN] a pointer indicate the structure which isused to store vrrp infomation when the function return.

Return

status_t

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

2.14 CMM

This section describes the macro, enum, struct and functions for trunk. They are defined in the header file “adlink_api_cmm.h”.

2.14.1 Macro Definitions

AD_IPMI_BUF_LEN.

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

The max ipmi message data length.

2.14.2 EnumType

None.

2.14.3 Data Structure

adIpmiReqT

Prototype

```
typedef struct {
    uint8_t     netFun;                     /* IPMI network function */
    uint8_t     cmd;                         /* IPMI command Id */
    uint8_t     reqData[AD_IPMI_BUF_LEN];   /* IPMI request data */
    uint8_t     reqLen;                     /* IPMI request data length */
} adIpmiReqT;
```

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

To load the ipmi request data.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

adlink_cmm_raw_send_req_t

Prototype

```
typedef struct adlink_cmm_raw_send_req_st {
    char    res_id_str[32];          // The name of the board, Ex: shelf, board_1, board_2....
    adIpmiReqT req;                  // ipmi request data
} adlink_cmm_raw_send_req_t;
```

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

To load the raw ipmi request message data.

adIpmiRspT

Prototype

```
typedef struct {
    uint8_t     netFun;                     /* IPMI network function */
    uint8_t     cmd;                        /* IPMI command Id */
    uint8_t     rspData[AD_IPMI_BUF_LEN];   /* IPMI response data */
    uint8_t     rspLen;                     /* IPMI request data length */
} adIpmiRspT;
```

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

To load the ipmi response data.

adlink_cmm_raw_send_rsp_t

Prototype

```
typedef struct adlink_cmm_raw_send_rsp_st {
    int             rc;                 // return 0 means success;
    adIpmiRspT    rsp;                  // ipmi response data
} adlink_cmm_raw_send_rsp_t;
```

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

To load the raw ipmi response message data.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

adlink_cmm_set_hotswap_state_req_t

Prototype

typedef struct adlink_cmm_set_hotswap_state_req_st {

uint8_t msg_type; // 0 – hotplug ready; 1 – hotplug request; 3 – hotplug success; 4 – hotplug failed

uint8_t cpu_node; // CPU node ID

} adlink_cmm_set_hotswap_state_req_t;

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

To tell CMM the hotplug state of CPU node.

adlink_cmm_get_hotswap_state_rsp_t

Prototype

typedef struct adlink_cmm_get_hotswap_state_rsp_st {

int rc; // return 0 – means success

uint8_t msg_type; // 0 – hotplug ready; 1 – hotplug request; 3 – hotplug success; 4 – hotplug failed

} adlink_cmm_get_hotswap_state_rsp_t;

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

To let CPU node read hotplug event detected by CMM.

adlink_cmm_reset_payload_req_t

Prototype

typedef struct adlink_cmm_get_hotswap_state_rsp_st {

char res_id_str[32]; // The name of the board, Ex: shelf, board_1, board_2....

uint8_t reset_type; // reset type: 0 - power down; 1 - power up; 2 - powercycle; 3 - hard reset; 5 - soft shutdown;

} adlink_cmm_get_hotswap_state_rsp_t;

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

To reset CPU node & COMe of switch.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

adlink_cmd_syserror_rsp_st

Prototype

```
typedef struct adlink_cmd_syserror_rsp_st{
    int rc;     // return 0 – means success;
} adlink_cmmd_syserror_rsp_t;
```

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

To general error return struct for CMM APIs.

adlink_cmm_get_payload_slot_rsp_t

Prototype

```
typedef struct adlink_cmm_get_payload_slot_rsp_st{
    int rc;                 // 0 means success;
    uint8_t  slot_count;    // the number of slots of this chassis
    uint16_t  use_mask;     // A bit mask to tell whether a slot is used. A bit is set to 1 if the corresponding slot is used. else wise it is cleared to 0
} adlink_cmm_get_payload_slot_rsp_t;
```

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

To tell how much CPU nodes are inserted, and tell in which slot the CPU board plug.

adlink_cmm_get_switch_number_rsp_t

Prototype

```
typedef struct adlink_cmm_get_switch_number_rsp_st{
    int rc;                 // 0 means success;
    uint8_t  sw_num;        // number of switch boards that plugin;
} adlink_cmm_get_switch_number_rsp_t;
```

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

To tell one or two switch board plugged in the Chassis.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

adlink_cmm_get_switch_cmm_state_rsp_t

Prototype

```
typedef struct adlink_cmm_get_switch_cmm_state_rsp_st{
    int rc;                 // 0 means success;
    uint8_t     slot_id;    // switch slot id:   0 or 1;
    uint8_t     state;      // 0 - CMM is backup, 1 - CMM is main;
} adlink_cmm_get_switch_cmm_state_rsp_t;
```

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

To tell current switch board slot id and its state.

adlink_cmm_get_id_rsp_t

Prototype

```
typedef struct adlink_cmm_get_id_rsp_st {
    int rc;                 // 0 means success;
    uint16_t    buf_len;    // chassis ID length of byte;
    uint8_t     buff[255];  // point of buf that stores the chassis ID string;
} adlink_cmm_get_chassis_id_rsp_t;
```

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

To read the chassis ID information.

2.14.4 Functions

adlink_sys_chassis_raw_send ()

Prototype

sadSysErrorT adlink_sys_chassis_raw_send(CLIENT* clnt, adlink_cmm_raw_send_req_t* req_data, adlink_cmm_raw_send_rsp_t* rsp_data)

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

To send raw ipmi command to CMM or CPU board.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Parameters

clnt [IN] RPC client that the init function returned.
req_data [IN] ipmi raw request data.
rsp_data [OUT] ipmi raw response data.

Return

sadSysErrorT

adlink_sys_chassis_set_rrc_hotswap_state ()

Prototype

adSysErrorT adlink_sys_chassis_set_rrc_hotswap_state(CLIENT* clnt,
adlink_cmm_set_hotplug_state_req_t* req_data)

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

To tell CMM the hotplug state of CPU board.

Parameters

clnt [IN] RPC client that the init function returned.
req_data [IN] ipmi set hotplug request data.

Return

sadSysErrorT

adlink_sys_chassis_get_rrc_hotswap_state ()

Prototype

adSysErrorT adlink_sys_chassis_set_rrc_hotswap_state(CLIENT* clnt,
adlink_cmm_get_hotswap_state_rsp_t* pRsp)

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

To get the hotplug state from CMM.

Parameters

clnt [IN] RPC client that the init function returned.
pRsp [OUT] ipmi get hotplug response data.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Return
sadSysErrorT

adlink_sys_chassis_reset_payload ()

Prototype
adSysErrorT adlink_sys_chassis_reset_payload (CLIENT* clnt,
adlink_cmm_reset_payload_req_t* req_data)

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description
To reset CPU board & Switch COME.

Parameters
clnt [IN] RPC client that the init function returned.
req_data [IN] ipmi reset payload request data.

Return
sadSysErrorT

adlink_sys_chassis_get_payload_slot_usemask ()

Prototype
adSysErrorT adlink_sys_chassis_get_payload_slot_usemask (CLIENT* clnt,
adlink_cmm_get_payload_slot_rsp_t* pRsp)

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description
To read information of plugged CPU board number, and slot id, etc.

Parameters
clnt [IN] RPC client that the init function returned.
pRsp [OUT] ipmi read chassis CPU board information response data.

Return
sadSysErrorT

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

adlink_sys_chassis_get_switch_number ()

Prototype

adSysErrorT adlink_sys_chassis_get_switch_number (CLIENT* clnt,
adlink_cmm_get_switch_number_rsp_t* pRsp)

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

To read information of switch board number that plugged in chassis.

Parameters

clnt [IN] RPC client that the init function returned.
pRsp [OUT] ipmi read chassis CPU board information response data.

Return

sadSysErrorT

adlink_sys_chassis_get_switch_cmm_state ()

Prototype

adSysErrorT adlink_sys_chassis_get_switch_cmm_state (CLIENT* clnt,
adlink_cmm_get_switch_cmm_state_rsp_t* pRsp)

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

To read onboard CMM state and slot id.

Parameters

clnt [IN] RPC client that the init function returned.
pRsp [OUT] ipmi read onboard CMM state and slot id information response data.

Return

sadSysErrorT

adlink_sys_chassis_get_id ()

Prototype

adSysErrorT adlink_sys_chassis_get_id (CLIENT* clnt, adlink_cmm_get_chassis_id_rsp_t*
pRsp)

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Supported on

aTCA-3710	aTCA-3430	aTCA-N700	cPCI-6S10	MXN-0410	MXN-3610/MXN-4100

Description

To read information of chassis identification, such like chassis serial number, etc.

Parameters

clnt [IN] RPC client that the init function returned.

pRsp [OUT] ipmi read chassis identification information response data.

Return

sadSysErrorT

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Getting Service

Ask an Expert: http://askanexpert.adlinktech.com

ADLINK Technology, Inc.
Address: 9F, No.166 Jian Yi Road, Zhonghe District
New Taipei City 235, Taiwan
新北市中和區建一路 166 號 9 樓
Tel: +886-2-8226-5877
Fax: +886-2-8226-5717
Email: service@adlinktech.com

Ampro ADLINK Technology, Inc.
Address: 5215 Hellyer Avenue, #110, San Jose, CA 95138, USA
Tel: +1-408-360-0200
Toll Free: +1-800-966-5200 (USA only)
Fax: +1-408-360-0222
Email: info@adlinktech.com

ADLINK Technology (China) Co., Ltd.
Address: 上海市浦东新区张江高科技园区芳春路 300 号 (201203)
300 Fang Chun Rd., Zhangjiang Hi-Tech Park, Pudong New Area
Shanghai, 201203 China
Tel: +86-21-5132-8988
Fax: +86-21-5132-3588
Email: market@adlinktech.com

ADLINK Technology GmbH
Address: Hans-Thoma-Strasse 11, D-68163, Mannheim, Germany
Tel: +49-621-43214-0
Fax: +49-621 43214-30
Email: emea@adlinktech.com

Please visit the Contact page at www.adlinktech.com for information on how to contact the ADLINK regional office nearest you.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000